Gestion automatisée de parcs informatiques : BpBatch

Par, Olivier Pagé o@esm2.imt-mrs.fr

École supérieure de Mécanique de Marseille (ESM2), Technopôle de Château Gombert, 13451 Marseille Cedex 20

Gérard Milhaud (g@esil.univ-mrs.fr),

École supérieure d'ingénieurs de Luminy (ESIL), Luminy - Case 925 - 13288 Marseille Cedex 9

pour le CISCAM (Centre Informatique et Systèmes de Communication d'Aix-Marseille) - *http://ciscam.univ-aix.fr*

Cet article est en ligne à http: //www.esm2.imt-mrs.fr/~olivier/bpbatchl.html

Résumé : Cet article à pour objet de décrire pas à pas l'installation et l'utilisation de bpbatch, logiciel gratuit mais non libre. Cet outil permet de « scripter » le comportement d'une machine lors de sa phase de démarrage. Différents exemples de scripts sont proposés, fournissant à l'administrateur du site les connaissances nécessaires à l'installation et la synchronisation d'un parc hétérogène de machines linux et Windows9x. Dans un premier temps, nous décrirons la création d'une image modèle de machine Linux, la configuration des serveurs DHCP et TFTP, puis nous étudierons la création de scripts permettant de déployer cette image. Dans un second temps, nous nous attarderons sur les subtilités nécessaires à l'utilisation de cette technique pour des machines Windows9X. Enfin, nous aborderons les limitations de cette technique et son couplage avec d'autres outils de déploiement.

Introduction

« Comment se rendre en toute quiétude à JRES 2001 pendant 3 jours, permettre à toute secrétaire de réinstaller elle-même sa machine, garantir que les Travaux Pratiques pourront se dérouler convenablement et ce en l'absence de toute compétence informatique sur place ? »

Pour tenter de répondre à cette problématique, plusieurs solutions ont été tour à tour testées (installation automatique kickstart/ZAK, diskless, terminaux ...), la plus satisfaisante à ce jour reposant sur l'utilisation d'un outil gratuit (mais non libre) developpé initialement par l'université de Genève : bpbatch.

Plus généralement, ce type d'outil vise à assister l'administrateur système confronté aux paradoxes récurrents suivants :

- le nombre de machines gérées et de services exigés croissent exponentiellement
- plus de souplesse dans l'hétérogénéité du parc, plus de qualité de service lui sont demandés
- les effectifs du service informatique sous sa responsabilité, quand ils ne sont pas nuls, restent stables et son temps de travail hebdomadaire tend à se réduire.

Tout ceci ne peut se régler sans banaliser à l'extrême les postes de travail sous sa responsabilité, supposant donc une centralisation des données utilisateurs.

Tout poste de travail doit pouvoir être reconstruit, échangé, à tout moment et ce sans perte de données.

Cet outil, par opposition à d'autres solutions existantes, ne nécessite aucun OS spécifique puisque reposant exclusivement sur TFTP et DHCP. Il s'intègrera donc très aisement dans le parc existant et diminuera très sensiblement le « coût total de possession » .

L'originalité de ce produit très simple est de fournir dès la phase de boot du système un interpréteur de commandes et donc de permettre de définir différents profils de comportement de la machine avant même le démarrage de l'OS.

Ainsi, la machine pourra procéder à un démarrage normal parmi un ensemble d'OS installés précédemment, une réinstallation totale de la machine, d'une ou plusieurs partitions, une modification mineure de quelques fichiers, le tout pouvant être conditionné par la saisie de password, de choix proposés aux utilisateurs etc...

Nous aborderons très succintement la mise en oeuvre des serveurs TFTP et DHCP pour nous attarder davantage sur la mise en oeuvre de différentes stratégies répondant à divers profils d'utilisateurs et d'utilisations.

Attention: bpbatch ne fonctionne que pour les partitions ext2 et fat (16 ou 32). L'exposé se concentrera donc sur l'administration de machines linux et/ou windows W9x.

Fonctionnement général

Pré-requis: disposer d'une carte réseau compatible PXE.(ftp://download.intel.com/ial/wfm/pxespec.pdf) 3com, Intel, IBM, Realtek en proposent, bootix (http://www.qualystem.com/prixpxe.html) et d'autres sociétés fournissent les ROM manquantes à vos cartes réseaux existantes.

Construction d'une image

1- Une machine modèle est (ré)installée et paramétrée.
2- L'administrateur crée une « image» compressée au travers de l'utilitaire fourni.
3- Cette image est ensuite déposée sur le serveur TFTP. (Comme nous le verrons, il est préférable de faire simultanément ces deux opérations au travers de NFS ou SMB).

Fonctionnement d'un client

1- La machine « bpbatch-isée » démarre. Elle boote au travers de sa carte réseau.
2- Elle obtient ses paramêtres réseaux ainsi que d'autres spécifiques à bpbatch au travers du serveur DHCP.
3- Elle charge dans un premier temps l'interpréteur de commandes bpbatch et le script l'accompagnant (TFTP).
4- Elle exécute ce script.

Dans sa forme la plus simple, nous supposerons que celui ci ne permette qu'une installation inconditionnelle de la machine (4 lignes à écrire correspondant aux étapes suivantes!!!):

Le script:

4.a- prend en charge le partionnement du disque.
4.b- puis récupère au travers de TFTP l' « image » sur le serveur qu'il copie localement (derrière la dernière partition définie).
4.c- décompresse cette image locale dans la partitition définie.
4.d- Démarre l'OS correspondant.

Comportement correspondant en production:

Si l'image n'est pas modifiée, cette machine redémarre (étapes 1 à 4), elle compare l'image locale (en cache) et l'image distante (serveur TFTP) et décompresse l'image locale dans la partition définie.
Ce fonctionnement permet un gain de temps et surtout de bande passante.

L'installation de machines Linux étant beaucoup plus simple que celle de machines windows, nous développerons dans un premier temps la gestion de ces premières.

Utilisation de bpbatch pour la gestion de machines Linux

Construction d'une image

Après avoir installé et paramétré la machine, récupérer bpbatch (http://www.bpbatch.org).
L'utilitaire de création de l'image compressée utilisé se nomme mrzip.
Créer un fichier texte avec pour extension .mrz (dans notre cas : mkimg.mrz)

```
showlog
set FragmentSize=«30 MB»
filter -«tmp/»
filter -«var/log»
filter -«/image»
fullzip «/» «/image/linux.imz»
```

lancer alors la commande

```
chemin/de/mrzip -b mkimg
```

(en omettant l'extension .mrz)

L'exécution de ce fichier permet la création d'une image nommée /image/linux.imz, correspondant à la partition /.

En fait, l'image /image/linux.imz est composée de n fichiers /image/linux.im* de 30 Mo (limite intrinsèque des serveurs tftp standards).

Cette image sera faite en omettant les répertoires tmp et var/log et bien sur /image ...
Dans un second temps, transférer ces fichiers linux.im* sur le serveur dans le repertoire accessible par tftp.

Il est beaucoup plus intéressant de monter par NFS (vérifier les droits d'écriture !!!) le repertoire /tftpboot du serveur. Création et transfert de l'image seront donc simultanés.

De plus, en copiant tous les binaires utilisés par bpbatch dans ce répertoire, nous n'aurons pas à l'installer sur tous les postes.

Pour notre exemple, nous le monterons en /tftpboot sur le client.

Le script s'écrit alors:

```
showlog
set FragmentSize=«30 MB»
filter -«tmp/»
filter -«var/log»
fullzip «/» «/tftpboot/linux.imz»
```

sur le client:

```
mount serveur:/tftpboot /tftpboot;cd /tftpboot;./mrzip -b mkimage
```

Configuration du serveur

Avant tout, nous devons écrire le script qu'exécutera le client au travers de bpbatch et qui permettra la réinstallation de la machine.
Ce script portera l'extension .bpb (ici linux.bpb).

```
hidelog
setpartitions «linux-ext2:2000 linux-swap:256»
setbootpart 1
fullunzip «linux.imz» 1
clean 2
linuxboot «{:1}/boot/vmlinuz» «root=/dev/hda1»
```

soient respectivement et très simplement:

- rester discret.
- créer la table des partitions avec une part. linux de 2000Mo et une part. swap de 256Mo.
- rendre la partition 1 bootable.
- récupérer l'image linux.imz si elle diffère de l'image en cache, la copier dans le cache et la décompresser dans la partition 1.
- booter le noyau vmlinuz situé dans le répertoire /boot de la partition 1 du disque 0 et passer comme argument root=/dev/hda1.

Le fonctionnement correspondant sera alors:

```
Client                                  Serveur

——— DHCP request ———->
<—— réponse DHCP : ———-
<——  Paramêtres IP ———-
<—— +nom du fichier de boot ——
<—— +nom du script ———-

——- TFTP request bpbatch ——>
<——— bpbatch  ———-

——- TFTP request script ——->
<——— linux.bpb ———
```

et si besoin (image différente de celle stockée en cache)

```
—— TFTP request linux.imz —>
<——— linux.imz ———
—— TFTP request linux.im1 —>
<——— linux.im1 ———
—— TFTP request linux.im2 —>
<——— linux.im2 ———
.....
```

Il nous faut donc installer un serveur TFTP ainsi qu'un serveur DHCP.

Serveur DHCP

Nous ne nous attarderons pas non plus sur l'installation d'un serveur DHCP (ici ISC 2.0).
La nécessité de gérer des machines présentant des différences matérielles et la volonté d'identifier à tout instant une machine nous a conduit à créer un serveur DHCP statique (une adresse MAC <-> une adresse IP).

Les paramètres nécessaires au fonctionnement de bpbatch sont :

```
filename «bpbatch»;
option option-135 «linux»;
option dhcp-class-identifier  «PXEClient»;
option vendor-encapsulated-options 01:04:00:00:00:00:ff;
```

soient respectivement:
- le fichier à charger via tftp au démarrage (l'interpréteur de commandes bpbatch).
- le nom du script à charger via tftp et à lancer au travers de bpbatch (ici « linux » sans l'extension .bpb)
- deux lignes ... spécifiques à bpbatch !....

D'autres options dans l'intervalle option-128 à option-134 peuvent également être récupérées par bpbatch ce qui nous permettra de préciser des caractéristiques matérielles. (c.f. plus loin)

En annexe, vous trouverez un exemple commenté de fichier dhcpd.conf.

Serveurs TFTP

Nous ne nous attarderons pas sur l'installation d'un serveur tftp standard.....Vérifier sa présence dans /etc/inetd.conf ou le lancer en standalone.

Pour des raisons de sécurité évidentes, il est préférable de limiter les requêtes au seul repertoire tftpboot. De même, limitez l'accès de l'extérieur à ce serveur tftp sur votre routeur et/ou au travers de tcp-wrapper.

A noter l'existence de serveurs TFTP « améliorés » (c.f site http://www.bpbatch.org):

- Linux enhanced TFTP server

« an enhanced TFTP server that can send packets of up to 1408 bytes instead of the standard 512 bytes.This is a great enhancement that you should use »

Avis confirmé par les tests menés. Un « must ».

Le serveur se lance alors comme suit (ça ne s'invente pas).
/usr/sbin/tftpd -d/tftpboot -c 32 -l /var/log/tftp.log -s 1408 59

- Security TFTP server

Permet une authentification des utilisateurs (/etc/passwd, NIS, shadow).

Non testé.

- Broadcast TFTP server

« a home-made *Broadcast variant of TFTP. Using this server, we were able to download images to 25 clients on a heavily loaded 10 Mb ethernet network at 6 Mb/s (it is more efficient than the regular TFTP because it does not need to acknowledge each packets) »*
Non testé.

Malheureusement, ces serveurs ne sont pas sous licence GPL et n'existent pour la plupart que sous la forme de binaires Linux et Solaris.

Informations complémentaires:

Selon le type de server tftp utilisé, l'interpréteur de commande « bpbatch » doit être renommé:

- Serveur TFTP standard: bpbatch
- Serveur TFTP enhanced (packets de 1408 octets sur le port 59) : bpbatch.P
- Serveur TFTP propriétaires Bootix ou Intel: bpbatch.B

Gestion des différences matérielles

Autant que possible, les machines se devront d'être très semblables !.....
Dans le cas contraire, le prix du Go étant très modique, créez une nouvelle image spécifique à ce nouvea groupe de machines....

Généralement, les différences majeures sont :

- couple carte vidéo/écran
- carte son

Ces différences ont un impact sur quelques fichiers (XF86Config, /etc/modules ou équivalent, /etc/hwconf sur mandrake et autres ...)

Une fois ces fichiers repérés, il suffit de les copier sur le serveur tftp et de mofifier simplement le script de la facon suivante:

```
...
fullunzip «linux.imz» 1
if «BOOTP-Option-133»==«matrox» copy « XF86Config.matrox » « {:1}/etc/X11/XF86Config »
if «BOOTP-Option-133»==«nvidia» copy « XF86Config.nvidia » « {:1}/etc/X11/XF86Config »
if «BOOTP-Option-134»==«maestro3» copy « modules.maestro3 » « {:1}/etc/modules »
if «BOOTP-Option-134»==«ess» copy « modules» « {:1}/etc/modules »
....
linuxboot «{:1}/boot/vmlinuz» «root=/dev/hda1»
```

Les option-133 et option-134 sont tout simplement des options transmises au travers de DHCP.

Bpbatch supporte les systèmes de fichiers ext2 et donc peut tout à fait récupérer un fichier par tftp et le copier dans l'arborescence de la partition ext2.

La syntaxe est alors « {disk:partition}/chemin/du/fichier »

Les partitions 0 et -1 désignent respectivement le MBR et le cache.

Définition de profils de machines Linux

On peut généralement définir 3 types de machines Linux ainsi que les comportements souhaités au démarrage:

- **machines libre accès:**
  Impossibilité de booter en local, réinstallation systématique et automatique de la machine, possibilité de ne pas réinstaller limitée à l'administrateur.

- ***machines personnelles:***
  Boot réseau systématique, choix de la réinstallation ou démarrage normal laissé à l'appréciation de l'utilisateur.

- **machines exceptionnelles:**
  Boot réseau possible, choix de la réinstallation ou démarrage normal laissé à l'appréciation de l'utilisateur.

A ces différents types de machines, nous associerons tout naturellement une option au travers de DHCP qui sera ensuite évaluée lors du script bpbatch (par exemple l'option-129).

Démarrage en local

Le problème du démarrage en local est très vite levé par la mise en place de password sur les BIOS des cartes mères et réseau mais aussi au travers de bpbatch:

ne pas copier le noyau lors de la création de l'image. Rajouter la ligne suivante dans le script .mrz
```
filter -«boot/»
```

modifier légèrement le script .bpb de la facon suivante :

```
suppression de setbootpart 1
linuxboot «vmlinuz» «root=/dev/hda1»
```

La machine n'a plus de noyau sur le disque dur mais le charge sur le serveur tftp afin de pouvoir booter.

Interface utilisateur

Offrir un choix à l'utilisateur est relativement aisé avec bpbatch qui dispose de directives graphiques et d'entrées-sorties :

```
hidelog
initgraph «800x600»
Drawgif «gifs/logo.gif»

:Tryagain
At 17,30 Print « (1)boot (2)reinstall :»
Waitforkey 60 goto boot
Getkey choice
if «$choice»== «1» goto boot
if «$choice»== «2» goto install
goto Tryagain

:install
setpartitions «linux-ext2:2000 linux-swap:256»
fullunzip «linux.imz» 1
clean 2

:boot
linuxboot «vmlinuz» «root=/dev/hda1»
```

Demander la saisie d'un password dans un temps donné (ici 60 secondes) est tout aussi simple:

```
At 15,30 Print « Entrez password admin ou taper sur <enter>: »
waitforkey 60 goto install
GetPasswd userpass 8
if «$userpass» Match-Passwd «dyOu6TzvoiF.I» goto boot
goto install
```

(ici le password est un copier-coller d'un password DES)

Systèmes Linux à plusieurs partitions

Modifier le script de création d'image comme suit (par exemple si /usr est la 2ème partition) :

```
fullzip «{:2}» «/tftpboot/usr.imz»
```

et non

```
fullzip «/usr» «/tftpboot/usr.imz»
```

Modifier le script d'installation comme suit:

```
setpartitions «linux-ext2:2000 linux-ext2:2000 linux-swap:256»
fullunzip «linux.imz» 1
fullunzip «usr.imz» 2
clean 3
```

Afin de tirer profit de la taille des disques existant, il est tout à fait possible de créer l'image d'une partition vide et de la mettre à disposition des utilisateurs. La durée de vie de celle-ci étant alors limitée par le prochain redémarrage. (partition scratch).

Conclusion sur l'utilisation de bpbatch pour les machines Linux

Comme nous venons de le voir, il est très facile de mettre en oeuvre différentes politiques de gestion du démarrage des machines. De plus, l'utilisation de bpbatch permet d'organiser des séances de formation d'administration système avancée, un simple redémarrage des machines permettant un retour au fonctionnement normal.

A chaque profil machine/utilisateur nous pouvons associer un script différent. Cependant, nous vous conseillons de définir un script unique pour toutes les machines et de tester la valeur des options supplémentaires transmises via DHCP.

(ici option-129 est le type de machine et option-130 le nom de l'image)

La directive include « filename » permet d'inclure un fichier au script.

Ainsi, les fichiers inclus correspondant à l'option-129 présentent une interface utilisateur différente et positionnent en retour la valeur du choix.

Les fichiers inclus correspondant à l'option-130 définissent la table de partitions et le nom de l'image à installer.

Un tel « squelette » de script serait alors:

```
initgraph "800x600"
hidelog
Drawgif "gifs/esm2_big.gif"

if "$BOOTP-Option-129"=="multiboot" include "include/multiboot.inc"
if "$BOOTP-Option-129"=="nopass" include "include/nopass.inc"
if "$BOOTP-Option-129"=="passwd" include "include/passwd.inc"

:install
if "$BOOTP-Option-130" == "w98" include "include/w98.inc"
if "$BOOTP-Option-130" == "debwin" include "include/debwin.inc"
if "$BOOTP-Option-130" == "deb" include "include/deb.inc"

:boot
if "$BOOTP-Option-130" == "w98" goto windows
include "include/kernel.inc"
linuxboot "{:1}/boot/$kernel" "root=/dev/hda1»

:windows
hidebootprom
if "$BOOTP-Option-129"=="multiboot" hdboot :3
hdboot :1
```

Utilisation de bpbatch pour la gestion de machines W9x (FATxx)

Côté client

Création d'une image

Récupérer le package bpbatch.zip (www.bpbatch.org), l'extraire et utiliser mrzip.exe
Rien de bien nouveau dans le script mkimg.imz:

```
      showlog
      filter -"windows/schedlog.txt"
      filter -"windows/msimgsiz.dat"
      filter -"*/index.dat"
      filter -"*.swp"
      filter -"*.tmp"
      filter -"temp/*"
      set FragmentSize="30 MB"
      fullzip "c:/" "h:/w98/w98.imz"
c:\>mrzip -b mkimg
```

Puis transférer cette image sur le serveur. De la même façon que précédemment, création et transfert peuvent être simultanés au travers de SMB (utilisation de samba). Si le fichier bpbatch.zip a été extrait dans le « partage » windows, vous n'aurez donc pas à l'installer sur tous les postes.

Création du script de boot (ici win98.bpb)

Version 1 (incomplet) :

```
hidelog
setpartitions "bigdos:1024"
setbootpart 1
fullunzip "w98/w98.imz" 1
hidebootprom
hdboot :1
```

soient respectivement :

- mode « discret » .
- création de la partition fat32 de 1Mo.
- définition de la partition bootable.
- récupération de l'image sur le serveur TFTP si différent de l'image en cache, décompression
- commande recommandée pour éviter tout problême de mémoire avant de lancer Windows
- booter sur la partition 1

Problème 1:
W9x ne sait pas configurer seul le hostname (au sens Netbios) au travers de DHCP.

Résolution:
Ce hostname figure dans 3 clefs de registres. Si les bonnes valeurs sont précisées dans un fichier c:\temp\patch.reg sur la machine , il suffit de lancer dans l'AUTOEXEC.BAT la commande suivante :

```
regedit /L:c:\windows\system.dat /R:c:\windows\user.dat c:\temp\patch.reg
```

Le fichier c:\temp\patch.reg correct doit donc être créé à la volée lors de l'installation de la machine.

Il suffit donc de créer un fichier squelette /tftpboot/patch.reg sur le serveur comme suit:

```
REGEDIT4
[HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\VxD\VNETSUP]
        "ComputerName"="${BOOTP-Host-Name}"

[HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\VxD\MSTCP]
        "HostName"="${BOOTP-Host-Name}"

[HKEY_LOCAL_MACHINE\System\CurrentControlSet\control\ComputerName\ComputerName]
        "ComputerName"="${BOOTP-Host-Name}"
```

Grâce à la directive patch, ${BOOTP-Host-Name} sera remplacée par sa valeur lors de l'éxécution de bpbatch puis copiée sur la machine dans c:\temp\patch.reg.

Le script devient:

Version 2 :

```
hidelog
setpartitions "bigdos:1024"
setbootpart 1
fullunzip "w98/w98.imz" 1
patch «patch.reg» «{:1}temp/patch.reg»
hidebootprom
hdboot :1
```

Probleme 2: Propres aux cartes 3C905C (semble-t-il ..)
la variable BOOTP-Host-Name n'est pas connue !!!

Résolution:

transmettre par DHCP au travers de l'option-132 (par exemple) le nom netbios de la machine, corriger en conséquence le fichier /tftpboot/patch.reg

Problème 3: Différences matérielles
Microsoft W9x a besoin d'installer les fichiers des drivers « un peu partout » sur le disque, dans la base de registres, etc. L'option retenue pour les machines linux n'est donc pas utilisable ici.
La solution la plus simple (si vous avez du disque) est de créer une image par type de machine !!!
Il reste cependant possible de faire autrement moyennant quelques litres de sueur.

L'astuce:
Microsoft W9x fonctionne (!) en plug-and-play : les nouveaux périphériques sont généralement détectés au démarrage. Si les drivers sont sur la machine, Microsoft W9x devrait les installer seul.
L'astuce consiste donc à installer sur chaque machine les fichiers nécessaires.

Détail des opérations:
- Copier l'ensemble du répertoire WIN98 du CD dans une partition vierge (D: par exemple)
- Copier les drivers supplémentaires nécessaires dans ce répertoire D:\WIN98
- Copier l'ensemble des fichiers *.inf de D:\WIN98 dans c:\windows\inf

Créer alors l'image du disque D:

Le script de création des images devient:

```
showlog
filter -"windows/schedlog.txt"
filter -"windows/msimgsiz.dat"
filter -"*/index.dat"
filter -"*.swp"
filter -"*.tmp"
filter -"temp/*"
set FragmentSize="30 MB"
fullzip "c:/" "h:/w98/w98.imz"
fullzip "d:/" "h:/w98/cd.imz"
```

Le script d'installation devient:

```
hidelog
setpartitions "bigdos:1024 bigdos:500"
setbootpart 1
fullunzip "w98/w98.imz" 1
fullunzip «w98/cd.imz» 2
patch «patch.reg» «{:1}temp/patch.reg»
hidebootprom
hdboot :1
```

Il reste un dernier problême (!!!):
Lorsque W9x installe un nouveau matériel, ce système à la mauvaise habitude de demander à redémarrer, ce qui aurait pour conséquence de réinstaller le système etc.

Pour contourner ce problème, il faut encore ruser finement :

1) Il est tout a fait possible de tester l'existence d'un fichier dans le script d'installation:
if exist "{:1}/runonce.bat" goto boot

Si ce fichier existe, ne pas installer la machine mais démarrer directement.
Ce fichier devra donc exister à l'issue de l'installation.
copy "runonce.bat" "{:1}/runonce.bat"

2) Il faut donc éviter que ce fichier n'existe encore à l'issue du premier boot complet (une fois les périphériques installés).

Pour ce faire, il est possible de forcer au travers de la base de registres (clef RunOnce) l'exécution d'un script au premier démarrage.

Ce script `(c:\runonce.bat)` aura pour seule fonction de s'effacer lui-même.

L'AUTOEXEC.BAT devra donc charger le fichier runonce.reg suivant :
`(regedit /L:c:\windows\system.dat /R:c:\windows\user.dat c:\temp\runonce.reg)`

```
REGEDIT4
[HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\RunOnce]
"zyva"="c:\\runonce.bat"
```

Conclusion sur l'utilisation de bpbatch pour les machines W9X

L'utilisation de bpbatch permet d'assurer un bon fonctionnement d'un parc de machines W9X. Les problèmes de virus, de dégradation de fonctionnement sont réglés par un simple redémarrage de la machine.
L'installation «sauvage» d'un logiciel sur les machines W9X se terminant généralement par une invite à redémarrer le système, celui ci aura donc disparu dans les minutes suivantes (effet de bord assez plaisant).

Conclusion générale et limitations

Mis en oeuvre sur plusieurs sites plus ou moins dépourvus totalement de personnel informaticien, cet outil a assuré une disponibilité élevée des équipements informatiques, tout dysfonctionnement d'une machine pouvant être éliminé par réinstallation automatique, et ce sans aucune compétence. La gestion des mises à jour et de la sécurité devient quasiment transparente. De plus, cette technique à parfaitement répondu aux contraintes de souplesse induites par les enseignements sur différents OS ainsi que la mise à disposition de machines « bac à sable » permettant l'enseignement des techniques d'administration système et réseau. De plus, cette application peut être couplée à d'autres outils de clonage (http://www.imagecast.com par exemple) en créant l'image de la disquette de boot créee par Imagecast et en l'installant sur le disque au travers de bpbatch.

Les limitations sont cependant nombreuses:

- bpbtach n'est pas sous licence GPL. Impossible de le faire évoluer.
- bpbatch est vieux.

(Les auteurs ont décidé de développer une version commerciale et délaissent cette application.)

- La somme des partitions créés ne peut excéder 8Go.
- Les partitions NTFS ne sont pas supportées.
- Le protocole reste TFTP (pas de multicast MTFTP).
- Rien n'est crypté.
- Impossibilité d'éditer les images pour modifications.
- Impossibilité de procéder à une installation « incrémentale ». Si 3 fichiers sont modifiés, l'image entière doit être chargée.

La version commerciale *(www.rembo.com)* résout bon nombre de ces limitations.
D'autres solutions libres sont à l'étude *(www.systemimager.org)*, *(replicator.sourceforge.net)* au moment de la rédaction.

Annexe - fichier dhcpd.conf

Bref rappel:

La syntaxe du fichier dhcpd.conf se compose de « paragraphes » imbriqués.

Cette syntaxe nous permet de regrouper par « type » les machines et de définir ainsi un ensemble de paramètres communs:

Ces paramètres peuvent bien sûr être surchargés dans un paragraphe inclus.

```
subnet  xx.yy.zz.0 netmask 255.255.255.0 {
#paramètres communs au réseau  :
        option domain-name-servers aa.bb.cc.dd, aa.bb.dd.cc;
        option domain-name «my.domain.not.com»;
        option subnet-mask 255.255.255.0;
        option broadcast-address xx.yy.zz.255;
        option routers xx.yy.zz.1
        default lease-time -1;
        max-lease-time -1;
        server-identifier xx.yy.zz.254; (en cas d'interfaces multiples ..)
        option nis-domain my.nis.domain;
#       etc....

        group yy{
        #Paramêtres spécifiques au groupe yy:
        #les options de bpbatch soient par exemple :
                filename «bpbatch»;
```

```
                option option-135 «mon-script-pour-yy»;
                option dhcp-class-identifier  «PXEClient»;
                option vendor-encapsulated-options 01:04:00:00:00:00:ff;
                option option-133 «matrox»;

                host a {
                #Définition du host a:
                        option host-name «a.my.domain.not.com»;
                        harware ethernet 00:01:02:F2:F3:F4;
                        option option-133 «nvidia»;
                }
                host b {
                #definition host b
                        option host-name «b.my.domain.not.com»;
                        harware ethernet 00:01:02:F3:F4:F5;
                }
        #fin group yy
        }

        group zz{
                ...
                host c {
                        ...
                }
        #fin group zz
        }
#fin subnet
}
```