

Mise en place de clients légers sous LTSP 5

François Ducrot, Université d'Angers



Plan

1. Présentation générale
2. Préparation du système de boot
3. Installation et paramétrage des fichiers exportés
4. Paramétrage du serveur interactif
5. Variantes éventuelles

I. Principe général

- Un client léger sans disque boote sur une distribution linux, chargée par le réseau
- Il lance une session interactive sur un serveur
- Toute la session se passe sur le serveur
- Le CL gère uniquement l'affichage et les périphériques
- Le système qui anime le CL est une distribution standard (fedora, debian, ubuntu, opensuse), modifiée légèrement grâce aux scripts fournis par LTSP
- LTSP apporte principalement :
 - un ensemble de script
 - un gestionnaire de connexions graphiques (LDM)
 - un système de fichiers pour les périphériques (ltspfs)

I. Répartitions des rôles

- Client léger : machine i386 ou x86_64 sans disque, RAM>256 Mo, carte réseau PXE
- Serveur interactif : c'est son CPU qui fait tout le travail. Pas mal de RAM
- Serveur de démarrage : fournit les fichiers nécessaires aux CL
 - DHCP : informations de démarrage
 - TFTP : noyau linux, initrd, et fichier de conf
 - NFS ou NBD : répertoire / complet
- Machine de préparation :
 - n'importe quelle machine i386 ou x86_64
 - utilisée uniquement pour la confection du système qui tournera ensuite sur les CL
 - distribution récente pour une bonne reconnaissance du matériel
- Réseau commuté à 100 méga

I. Séquence de boot

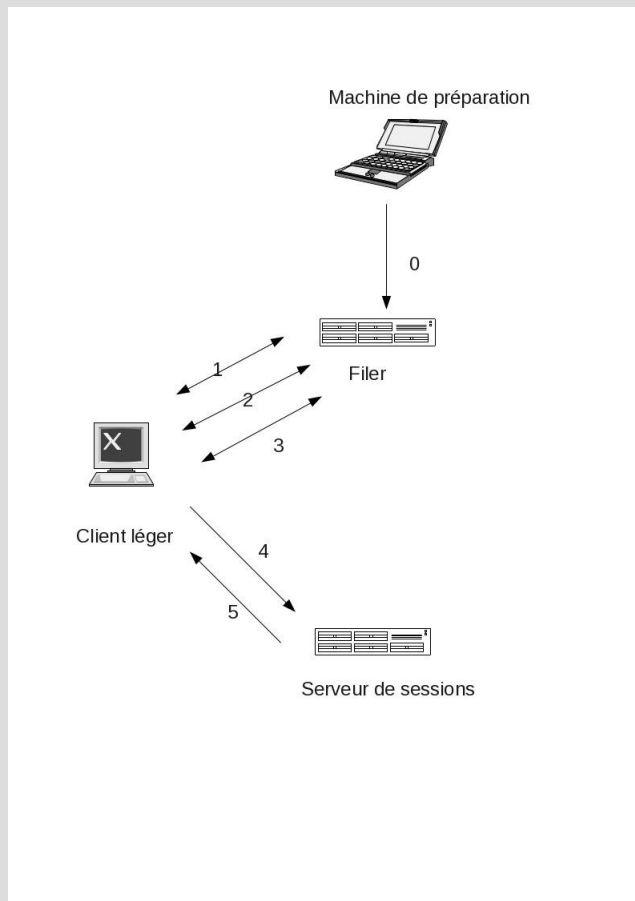
- Requête DHCP : adresse IP, nom des fichiers à télécharger
- Requête TFTP : téléchargement du noyau, d'un initrd et d'un fichier lts.conf
- Le CL boote sur l'initrd, reconfigure sa carte réseau
- Montage par NFS ou NBD d'une image de /
- Le CL effectue un chroot sur le nouveau /
- Le CL lance un gestionnaire de connexion LDM
- L'utilisateur se connecte au serveur de sessions interactives

Que fait LDM ?

- LDM lance depuis le CL une connexion ssh sur le serveur de sessions. Voici ce que donne `ps -edf` sur le CL :

```
ssh -Y -t -S /var/run/ldm_socket_3477_172.19.45.119 \
-l test 172.19.45.119 LTSP_CLIENT=172.19.45.176 \
LTSP_CLIENT_HOSTNAME=ltsp176 LC_ALL=fr_FR.utf8 \
LANGUAGE=fr_FR.utf8 LANG=fr_FR.utf8 \
DISPLAY=172.19.45.176:3 \
PULSE_SERVER=tcp:172.19.45.176:4713 \
ESPEAKER=172.19.45.176:16001 LTSP_ALSA_DEFAULT=pulse \
/etc/X11/Xsession default < /dev/null > /dev/null ; \
/usr/sbin/ltspfsmounter all cleanup
```

En résumé



0 : Installation des fichiers

1 : DHCP

2 : TFTP

3 : NBD

4 : Demande de connexion LDM

5 : Session interactive

Machine de préparation : Ubuntu 10.10

Filer : CentOS (ou Fedora)

Serveur de sessions : Fedora 14

II. Préparation du système

Tout le travail se fait sur la machine de préparation (ici : Ubuntu Maverick)

- Installation du paquet ltsp-server

```
apt-get install ltsp-server
```

- Création d'un chroot

```
ltsp-build-client --arch=i386 --chroot=demo --dist=maverick
```

- Utilise les utilitaires de la distribution (debootstrap) pour créer un système dans le répertoire /opt/ltsp/demo qu'on référencera par \$CHROOT
- Fabrique un répertoire /var/lib/tftpboot/ltsp/demo, contenant les fichiers à télécharger par TFTP
- Crée une image squashfs /opt/ltsp/images/demo.img, que la machine exporte par NBD
- A ce stade, on pourrait utiliser cette machine de préparation pour faire démarrer un CL, et le faire ouvrir une session sur cette même machine.
- Reste à affiner les choses...

II. Modification de l'image /

Installer les clés RSA des serveurs de sessions

Pour que LDM puisse se connecter sur le (ou les) serveur(s) de session, il faut que le CL connaisse leurs clés RSA. On va donc mettre ces clés dans le le volume / monté par le client.

- Récupérer les clés RSA des différents serveurs de sessions
- Les mettre dans un fichier `$CHROOT/etc/ssh/ssh_known_hosts`

```
# cat $CHROOT/etc/ssh/ssh_known_hosts  
  
interactif1.math ssh-rsa AAAAB3NzaC1yc2EAAAABIwAAAQEA1y3T...  
interactif2.math ssh-rsa AAAAB3NzaC1yc2EAAAABIwAAAQEA7pAo....
```

- Refabriquer une image `/opt/ltsp/images/demo.img` de `$CHROOT` :

```
ltsp-update-image -a demo
```

III. Travail sur le serveur de boot

Maintenant que l'image est prête, il reste à configurer le serveur de boot :

- Déclarer les CL dans le DHCP
- Installer et configurer les fichiers chargés par TFTP
- Activer l'export de l'image par NBD

III.1 DHCP

Rien de bien original :

```
#cat /etc/dhcp/dhcpd.conf

...

host CL1 {
    next-server fileserver.math;
    filename "/ltsp/demo/pxelinux.0";
    hardware ethernet 00:25:64:e0:75:a0;
    fixed-address 192.168.1.25
}

...
```

III.2 TFTP

- Copier le répertoire `/var/lib/tftpboot/ltsp/demo` de la machine de préparation sur au même endroit sur le serveur de fichiers

```
#ls /var/lib/tftpboot/ltsp/demo
abi-2.6.35-24-generic          nbi.img-2.6.35-24-generic
config-2.6.35-24-generic     pxelinux.0
gpxelinux.0                  pxelinux.cfg
initrd.img                   System.map-2.6.35-24-generic
initrd.img-2.6.35-24-generic  vmcoreinfo-2.6.35-24-generic
initrd.img-2.6.35-24-generic.ori  vmlinuz
lts.conf                     vmlinuz-2.6.35-24-generic
nbi.img
```

- Deux fichiers à regarder : `lts.conf` et `pxelinux.cfg/default`

III.2 TFTP(2)

```
#cat pxelinux.cfg/default  
  
default ltsp  
  
label ltsp  
kernel vmlinuz  
append ro initrd=initrd.img quiet nbdport=2000
```

Le noyau va monter le / par nbd sur le port 2000 du serveur de fichiers.

III.2 TFTP (3)

- Editer le fichier `lts.conf` pour décrire le comportement de chaque CL :

```
[default]
  LOCALDEV=True
  SOUND=True
  NBD_SWAP=False
  SCREEN_02=shell
  SCREEN_03=ldm
  LDM_DIRECTX=yes
  LDM_SERVER="interactif1.math interactif2.math"
  LOCAL_APPS=False
  X_NUMLOCK=True

[78:e7:d1:b5:a9:0b]
  VOLUME=30
  XKBLAYOUT="fr"
[78:e7:d1:b4:a2:35]
  VOLUME=20
  XKBLAYOUT="us"
```

- Pour toutes les options : `man lts.conf`

III.3 NBD

- Installer le logiciel `nbd-server`
 - Le paquet `nbd` existe sous Fedora, mais ne fonctionne pas comme attendu
 - Le paquet n'existe pas sous CentOS
- Le plus simple est donc de recompiler les sources `nbd-2.9.15.tar.bz2`
- Déposer l'image `demo.img` quelque part sur le serveur de fichier
 - Créer un fichier de config `/etc/nbd-server/config` pour `nbd`

```
[generic]
    user = root
    group = root

[demo]
    exportname = /exports/ltsp/demo.img
    port = 2000
    readonly = true
```

- Créer un script de lancement standard `/etc/init.d/nbd-server`

IV. Serveur interactif

- Pas besoin de XDM, GDM, KDM
- Veiller à ce que les paquets suivants soient bien installés :
 - `ltspfs` (montage de fichiers par `fuse`)
 - `xorg-x11-xauth` (pour l'authentification des sessions interactives)
- Un petit problème de compatibilité Ubuntu/Fedora :
LDM cherche un fichier `/etc/X11/Xsession`, qui n'est pas placé là sous fedora

```
ln -s /etc/X11/xinit/Xsession /etc/X11
```

Et c'est tout !

V. Variantes

a. Répartition de charge

- On veut que les clients choisissent aléatoirement le serveur interactif
- Créer un fichier `$CHROOT/usr/share/ltsp/get_hosts` contenant :

```
#!/bin/bash
TMP_LIST=""
SHUFFLED_LIST=""
for i in $SERVER_LIST; do
rank=$RANDOM
let "rank %= 100"
TMP_LIST="$TMP_LIST\n${rank}_$i"
done
TMP_LIST=$(echo -e $TMP_LIST | sort)
for i in $TMP_LIST; do
SHUFFLED_LIST="$SHUFFLED_LIST $(echo $i | cut -d_ -f2)"
done
echo $SHUFFLED_LIST
```

V-a Répartition de charge (suite)

- Refabriquer une image par `lts-sp-update-image` , puis la remettre sur le serveur de fichiers
- Modifier le fichier `lts.conf` en remplaçant la ligne

```
LDM_SERVER="interactif1.math interactif2.math"
```

Par

```
SERVER_LIST="interactif1.math interactif2.math"
```

V. Variantes

b. Serveur ssh sur le CL

- On veut installer un serveur ssh sur les CL, permettant par exemple de les éteindre à distance
- Installer `openssh-server` dans le chroot :

```
chroot /opt/ltsp/demo
-> mount -t proc proc proc
-> apt-get install openssh-server
-> exit
```

- Créer un couple clé privée/clé publique `demo.priv/demo.pub` sans mot de passe par `ssh-keygen`
- Copier `demo.pub` en `$CHROOT/etc/ssh/authorized_keys`
- Modifier `$CHROOT/etc/rc.local` en rajoutant :

```
if ! [ -d /root/.ssh ]; then
    mkdir /root/.ssh
fi
cp /etc/ssh/authorized_keys /root/.ssh
```

V-b. Serveur ssh sur le CL (suite)

- Activer l'usage du rc.local sur le CL

```
cd $CHROOT/etc/rc2.d  
ln -s ../init.d/rc.local S99rc.local
```

Remarque : l'utilisation de la copie de la clé programmée par rc.local au démarrage du CL est nécessaire, car le répertoire /root n'existe qu'en RAM (pas dans le / exporté)

- Refabriquer une image de \$CHROOT et l'installer sur le serveur
- Grâce à ce qui a été fait, on pourra éteindre le terminal à distance par :

```
ssh -o StrictHostKeyChecking=no -i demo.priv -f root@CL "poweroff"
```

V. Variantes

c. Utilisation d'un client léger comme kiosk

D'autres modes de construction d'un CL sont possibles

- Mode kiosk : le CL propose un navigateur web fonctionnant en local
- Client lourd (fat client)
- Client léger avec certaines applications locales

Expliquons juste la fabrication d'un kiosk

- Créer un nouveau chroot /opt/ltsp/kiosk , contenant les utilitaires nécessaire à l'utilisation en mode kiosk (firefox) :

```
ltsp-build-client --arch=i386 --chroot=kiosk --dist=maverick --kiosk
```

- Rajouter dans \$CHROOT/usr/share/ltsp le fichier kioskSession , qui manque dans la distribution :

```
# cat kioskSession

#!/bin/sh

KIOSKHOME=/usr/local/share/ltspkiosk/home
KIOSKSTARTUP=/usr/local/share/ltspkiosk/startup
TMPDIR=$(getent passwd ltspkiosk | cut -d: -f6)
export HOME=${TMPDIR}

if [ -d "${KIOSKHOME}" ]; then
    cp -r ${KIOSKHOME}/* ${KIOSKHOME}/.??* ${TMPDIR}
fi

[ -n "${XAUTHORITY}" ] && cp -a ${XAUTHORITY} ${TMPDIR}

for i in ${KIOSKSTARTUP}/* ; do
    [ -x "${i}" ] && eval "${i}" &
done

/usr/bin/firefox || ldm-dialog --message "Could not start program."

[ "$USER" != "root" ] && pkill -u $USER
exit 0
```

V-c Le mode kiosk (suite)

- Recréer une image `kiosk.img` du chroot `parltsp-update-image`
- Rajouter une entrée dans la config de `nbd` :

```
[kiosk]
  exportname = /exports/ltsp/kiosk.img
  port = 2001
  readonly = true
```

- Relancer `nbd-server`
- Copier `$TFTPBOOT/ltsp/demo` en `$TFTPBOOT/ltsp/kiosk`
- Modifier `$TFTPBOOT/ltsp/kiosk/pxelinux.cfg/default`

```
append ro initrd=initrd.img quiet nbdport=2001
```

- Modifier en conséquence l'enregistrement dans le `dhcpd.conf`