

Serveur de client léger LTSP v5

Emmanuel Le Normand
DSI université Pierre Mendès France
Rue des universités
38040 Grenoble

Résumé

Depuis la fin des années 90, le projet LTSP a pour but de recycler d'anciens PC en clients légers. Les premières versions n'étaient pas toujours faciles à mettre en œuvre mais LTSP v5 apporte un ensemble de fonctionnalités intéressantes permettant de gérer simplement un parc plus ou moins important de clients. Par exemple, le module «cluster» permet un équilibrage de charge et une redondance au niveau des serveurs d'application.

Actuellement environ 150 clients LTSP fonctionnent à l'université Pierre Mendès France (Grenoble 2) sous différentes configurations permettant de répondre aux divers besoins des usagers comme des bornes internet, une salle pour l'enseignement des langues, des postes en libre accès en différents endroits de l'université, le tout géré directement depuis les serveurs LTSP sans intervention sur les clients.

Bien adapté aux besoins des bibliothèques, LTSP est implanté sur l'ensemble des postes en libre service de la bibliothèque Droit/Lettres et celles du réseau SICD2. Nous travaillons également avec les autres bibliothèques universitaires grenobloises et de Savoie qui déploient aussi LTSP, dans le but d'avoir des environnements et procédures communes sur les postes publics.

Mots clefs

LTSP, clients légers, gestion de parc, poste public, administration système

1 Introduction

Le serveur de client léger LTSP v5 permet très simplement de créer un réseau de clients légers indépendants et configurables très facilement, en recyclant d'anciens PC en clients légers. Le recyclage du matériel permet de faire des économies importantes sur les budgets d'investissement mais LTSP fait économiser également le temps de maintenance alloué pour la gestion de parc.

L'installation du serveur LTSP ne présente pas de difficultés particulières si vous utilisez une distribution GNU/Linux classique comme Debian, Fedora, OpenSuse ou Ubuntu. Pour les distributions ne disposant pas de paquet LTSP, il est possible de récupérer les codes sources et de les compiler. Pour Fedora, reportez vous au projet «K12Linux», pour OpenSuse au projet «Kiwi-LTSP» et son interface «EasyLTSP». Le code source est disponible sur le site «launchpad.net».

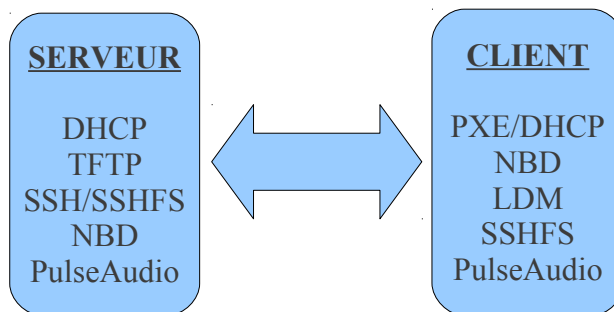
Les exemples dans cet article sont donnés pour une installation sous Ubuntu. Pour Debian, ce sont les mêmes actions, il existe juste quelques différences qui seront données au cas par cas. Cet article n'est pas écrit pour être lu de manière linéaire, mais par bloc pour donner une information globale vous donnant un aperçu de LTSP v5, en fonction de ce dont vous pourriez avoir besoin lors de la conception de votre réseau de clients légers.

2 Installation

Comme base de départ, on utilise un serveur Ubuntu x86_64 à jour avec seulement Openssh-server installé. Les installations et paramétrages peuvent être réalisés via un accès ssh sur le serveur. Le serveur peut être virtuel ou physique.

Pour installer LTSP sur votre serveur il faut, via votre outil de gestion de paquets, choisir «ltsp-server». Le paquet «ltsp-server-standalone» permet d'installer en plus le serveur DHCP. D'autres services seront installés : un serveur TFTP, un serveur NBD (ou pour Debian le serveur NFS), le serveur OpenBSD-Inetd et les outils pour la gestion PXE.

3 Schéma de principe



4 Environnement LTSP

L'«environnement LTSP» désigne l'ensemble des paquets ou applications qui seront utilisés ou utilisables par les ressources du client léger. Il y a trois modes d'environnement possibles sur LTSP ; classique, local-app ou autonome.

- Le mode «classique» est celui par défaut, c'est un environnement minimaliste qui ne contient que les fichiers ou exécutables nécessaires au fonctionnement du client, exemple le client NBD (ou NFS pour Debian), ou la partie du serveur X gérant le clavier et la souris,... Dans ce mode, toutes les applications (bureautique, navigateur,...) fonctionnent sur le serveur.
- Le deuxième mode de fonctionnement consiste à améliorer le mode «classique» en installant dans l'environnement des applications complémentaires qui tourneront sur le client pour alléger le serveur. C'est ce qu'on appelle le mode «local-app». Il permet de diminuer les besoins en ressources du serveur.
- Le dernier cas d'environnement que j'appelle «autonome», les anglophones utilisant le terme «fat-client», consiste à faire fonctionner l'ensemble des applications sur le client. Seules les connexions type SSH, NBD,..., sont visibles sur le serveur. C'est le cas le plus intéressant car les ressources du serveur peuvent être plus faibles tout en ayant beaucoup de clients connectés.

5 Caractéristiques matérielles pour un réseau LTSP

Les règles suivantes ne sont données qu'à titre d'exemple, il vous faudra les affiner en fonction de votre utilisation de LTSP.

5.1 Les caractéristiques matérielles pour un serveur LTSP

Le choix du serveur se fait surtout en fonction du type d'environnement utilisé (classique, local-app ou autonome). La règle habituelle de calcul pour des clients classiques est que le serveur doit avoir au minimum 512Mo de mémoire pour ses besoins propres plus 128 à 256Mo par client qui se connecteront.

Pour des clients autonomes, les besoins en ressources mémoires seront moins importants. Un exemple, pour 80 clients autonomes connectés, les ressources mémoires globales utilisées sont inférieures à 2Go.

Les autres caractéristiques du serveur : le processeur, double cœurs à quatre cœurs en fonction de l'usage et du nombre de client ; le réseau, une à plusieurs cartes giga-ethernet ; les disques, en fonction du système RAID choisi.

5.2 Les caractéristiques matérielles pour des clients classiques

Pour faire un client LTSP classique, on peut recycler d'anciennes machines ou utiliser des vrais clients légers type HP t5xxx, ayant comme caractéristiques minimales, un processeur x86 type Pentium 3, une mémoire RAM de 256Mo, et une carte réseau 100Mb compatible PXE. Pour le reste, carte graphique, carte son, lecteur CD ou autres doivent être standards. Aucun disque dur n'est nécessaire.

5.3 Les caractéristiques matérielles pour les clients autonomes

Pour les clients LTSP autonomes, on peut aussi recycler d'anciennes machines ou avoir de vrais clients légers type HP t5xxx, ayant comme caractéristiques minimales, un processeur x86 type Pentium 4, 1Go de RAM et une carte ethernet 100Mb compatible PXE. Pour le reste, il suffit de composants standards.

Mais si vous réalisez une borne internet pour la consultation de pages web basiques, les ressources nécessaires pour le client seront moindres, un Pentium 3 avec 512Mo de RAM et une carte Ethernet 100Mb compatible PXE suffiront.

6 Création du premier environnement LTSP

La création d'un environnement LTSP est très facile. On utilise la commande «ltsp-build-client» suivie des options que l'on veut activer dans cet environnement. Pour voir l'ensemble des options possibles utilisez :

```
ltsp-build-client --extra-help
```

Voici les options les plus couramment utilisées :

- --dist : permet de choisir une version d'Ubuntu (ou Debian) différente de la version serveur. Intéressant pour avoir le serveur avec une version stable et un environnement client utilisant les dernières versions de logiciel. (exemple --dist natty pour Ubuntu ou --dist squeeze pour Debian)
- --mirror : permet de spécifier un miroir d'installation différent. (exemple --mirror <http://mon.miroir.fr/mirror>)
- --arch : permet de spécifier le type de processeur des clients. Attention, si votre serveur est installé en 64bit, et que les clients sont en 32bit, il faut utiliser cette option pour créer l'environnement pour des x86 (exemple --arch i386). En théorie, pour créer des environnements pour PowerPC il suffirait de mettre l'option à «--arch PPC» mais cela ne fonctionne pas car il y a trop d'incompatibilités entre la compilation pour x86 et PPC. Si vous devez utiliser des clients PPC, mieux vaut créer votre environnement sur une machine PPC.
- --chroot : permet de donner un nom à votre environnement sinon par défaut c'est le type «arch» qui est pris. (exemple --chroot catalogue)
- --fat-client : permet de créer un environnement autonome. Cette option va se référer à ce qui est installé sur le serveur pour choisir le gestionnaire de bureau et des applications à installer. Cette option est spécifique à Ubuntu.
- --fat-client-desktop : permet de créer un environnement autonome. Contrairement à l'option «fat-client» citée au dessus, cette option permet de spécifier le nom du gestionnaire de bureau via les noms sous Ubuntu (exemple --fat-client-desktop kubuntu-desktop)
- --prompt-rootpass : permet d'activer le compte «root» et de lui donner un mot de passe. Ce compte root n'est actif que sur les clients, ce n'est pas le compte root du serveur. Cette option est

très utile pour déboguer des problèmes sur les clients (problème de reconnaissance matériel, droit d'accès,...). Lors de la création de l'environnement, il vous sera demandé de taper le mot de passe pour root.

- `--purge-chroot` : permet d'effacer les dossiers d'un environnement existant que l'on veut recréer avec le même nom.

Pour les autres options, reportez vous aux pages du manuel de la commande. Il est aussi possible de créer ses propres options pour la création d'environnement.

Les options peuvent différer d'une distribution GNU/Linux à l'autre. Exemple l'option `--mythbuntu` n'est valable que pour Ubuntu. Autre exemple, sous Debian on a l'option `--squashfs-image` qui va compresser une image de l'environnement pour NBD qui n'est pas utilisée par défaut.

Pour un environnement classique, il suffit de spécifier le nom du chroot et de l'architecture des clients.

```
ltsp-build-client --arch i386 --chroot classique --prompt-rootpass
```

Pour un environnement autonome, il faut spécifier l'option «fat-client» ou «fat-client-desktop» :

```
ltsp-build-client --arch i386 --chroot autonome --prompt-rootpass --fat-client-desktop ubuntu-desktop
```

7 La maintenance des environnements LTSP

Pour faire la maintenance du ou des environnements LTSP, il faut utiliser «chroot» qui vous permettra de vous placer dans l'environnement comme s'il était le système «racine». Ensuite il suffit de faire les modifications ou installations comme sur un système standard. Pour certaines des actions que vous voudrez mener, il faudra monter dans le système «chrooté», «/proc» et «/dev/pts».

En fonction de la distribution ou de la version de LTSP, vous aurez à faire les actions, de «chroot» et montages, soit manuellement soit avec l'utilitaire «ltsp-chroot» disponible pour le faire à votre place. La syntaxe est simple, mais il y a une erreur à ne pas commettre, l'option «--arch» ne correspond pas à celle du «ltsp-build-client». Ici il s'agit du nom de l'environnement. Par exemple :

```
ltsp-chroot --mount-dev-pts --mount-proc --arch demo
```

Cela montera «/proc et /dev/pts» et vous positionnera dans l'environnement «demo». En quittant l'environnement les points de montage «/proc et /dev/pts» seront démontés automatiquement.

8 Les autres outils de LTSP

En plus de la commande «ltsp-build-client», un ensemble d'outils est présent pour faire la maintenance des environnements.

- «ltsp-info» permet d'avoir des informations sur : la version de LTSP d'installée, les différents environnements présents, la version du serveur LDM et la localisation des fichiers liés aux environnements (dossier TFTP, fichier lts.conf,...)
- «ltsp-update-image» permet de recompresser les images après modification des environnements. Cela n'est utile que si vous utilisez NBD. Pour Debian vous n'avez pas à le faire par défaut. Exemple :

```
ltsp-update-image -a demo -p 2004
```

Cette commande va recompresser l'environnement «demo», positionner le port TCP de connexion à 2004 et modifier le fichier «/ltsp/demo/pxelinux.cfg/default» avec les options de démarrage nécessaires aux clients.

- «ltsp-update-sshkeys» est utile quand on change les clés SSH du ou des serveurs LTSP. Ce module va recopier les clés dans les dossiers /etc/ssh/ de chaque environnement.

- «Ltsp-update-kernel» est utilisé quand vous avez fait des mises à jour dans les environnements et que les noyaux ont été modifiés. On doit, dans ce cas, mettre à jour ceux qui sont situés dans les dossiers TFTP.

9 Le serveur DHCP

Pour pouvoir utiliser LTSP, il faut configurer le serveur DHCP pour répondre aux requêtes des clients PXE. Il est possible de se passer du serveur en configurant le démarrage sur des supports amovibles ou sur le disque dur de chaque client avec les options nécessaires, mais cela deviendrait très statique, empêchant les évolutions rapides.

Le service DHCP n'est pas à installer obligatoirement sur le serveur LTSP. Il peut l'être également sur un serveur Microsoft® Windows®.

Le serveur DHCP a besoin de connaître au moins deux informations à transmettre au client, si le serveur LTSP utilise NBD et une option supplémentaire s'il utilise NFS:

- option «filename» : c'est l'information donnant l'emplacement du fichier de démarrage initial (pour PXE : pxelinux.0) mais aussi le dossier de référence pour les téléchargements à venir (exemple : /ltsp/demo/pxelinux.0, redirige sur le dossier «/var/lib/tftpboot/ltsp/demo/» du serveur LTSP).
- option «nextserver» : c'est l'adresse IP du serveur TFTP, attention pas de nom DNS car le client PXE ne sait pas faire de résolution de nom.
- option «root-path» : n'est utile et obligatoire que pour une connexion NFS. Cette information est utilisée par le client NFS pour savoir où se trouve la racine de son système (exemple «/opt/ltsp/i386»).

10 NBD ou NFS ?

C'est la principale différence qui existe entre l'utilisation de LTSP sous Debian ou Ubuntu. Chacun de ces deux services a ses avantages et ses inconvénients. L'avantage principal de NFS est que chaque modification est immédiatement disponible sur les clients. Alors que pour NBD, il faut recompresser l'image ce qui peut perturber le fonctionnement des clients. L'avantage de NBD souvent mis en avant, est que les connexions entre le serveur et les clients sont plus sécurisées. Actuellement, pour utiliser le mode «autonome» de LTSP, on ne peut le faire facilement que sous Ubuntu avec NBD.

11 Le fichier lts.conf

C'est le fichier de paramétrage des clients LTSP. Il est situé dans le dossier TFTP de l'environnement (exemple : /var/lib/tftpboot/ltsp/demo/lts.conf). Par défaut il n'est pas créé, vous devrez le faire pour activer ou non des options. C'est un fichier texte qui se décompose en une section «default» pour définir les options par défaut de tous les clients, puis vous pouvez spécifier des options pour chaque machine. Elles sont de deux types, booléen (true ou false) ou à valeur fixe (chaîne de caractères).

```
#commentaire
[default]
    SERVER=192.168.0.100
    X_COLOR_DEPTH=16
    LOCALDEV=True
    SOUND=True
    NBD_SWAP=True
    SYSLOG_HOST=192.168.0.50
    XKBLAYOUT=fr
```

```
#on peut faire des groupes
[UN_GROUPE]
    SOUND=False
#spécificité pour une machine avec l'adresse ethernet
[11:22:33:44:55:66]
    LIKE=UN_GROUPE
    LOCALDEV=False
```

Les options standards que l'on trouve souvent dans la section «default», sont :

- **SERVER** : permet de spécifier l'adresse IP du serveur LTSP, utile lorsque l'on utilise un service TFTP qui n'est pas sur le serveur LTSP.
- **LOCALDEV** : active ou non l'utilisation des périphériques du client. Pour utiliser les clés USB il faudra activer cette option.
- **NBD_SWAP** : permet d'activer le SWAP par réseau via NBD. Vu que les clients n'ont pas de disque dur ils n'ont pas de partition swap.
- **XKBLAYOUT** : fixe le type de clavier, par défaut c'est le clavier anglais. Cette option doit être positionnée «fr» pour un clavier azerty français.

L'ensemble des options possibles se trouve expliqué dans la page «man» de lts.conf.

12 Environnement «local-app»

Pour utiliser ce type d'environnement, il faut d'abord créer un environnement LTSP «classique». Puis, en utilisant la procédure de maintenance vue plus haut, il suffit d'installer les applications que l'on veut utiliser localement. L'activation de la fonction «local-app» sera autorisée dans le fichier de configuration «lts.conf». On peut lancer, depuis le client, les applications en faisant précéder la commande exécutable de «lts-localapps» . Par exemple :

```
lts-localapps firefox %U
```

L'activation et le paramétrage des options «local-app» se font dans le fichier «lts.conf».

```
#activation de l'option local-app
LOCAL_APPS=True
#activation de la modification des lanceurs dans le menu
LOCAL_APPS_MENU=True
#liste d'applications devant être lancées en mode local-app
LOCAL_APPS_MENU_ITEMS=firefox,vlc
```

13 LTSP Display Manager (LDM)

LDM est le gestionnaire de session de LTSP. Il connecte les utilisateurs des clients légers au serveur LTSP au travers d'une connexion SSH. La connexion est ainsi sécurisée contrairement à XDMCP. Il est utilisé dans tous les modes d'environnement, avec en plus pour le mode autonome, la gestion du montage du dossier personnel (home directory) via le protocole SSHFS. Comme l'authentification s'appuie sur SSH, on peut facilement paramétrer les modules «pam.d» et ainsi utiliser un annuaire LDAP ou un contrôleur de domaine AD pour la gestion des utilisateurs.

Au niveau du fichier de configuration «lts.conf» on dispose aussi d'options spécifiques à LDM. Par exemple, on peut autoriser ou non l'utilisation d'un client à tel ou tel utilisateur.

```
[11:22:33:44:55:66]
```

```
LDM_ALLOW_USER=edge,dupond,tintin
```

On peut également se passer de LDM. Par exemple, si on construit une borne web sans identification en mode «autonome», il suffit d'installer un gestionnaire de session dans l'environnement puis de paramétrer l'autologin avec un compte générique n'existant que dans l'environnement.

14 TFTP et PXE

Les fichiers de boot des clients sont tous situés dans le dossier «/var/lib/tftpboot/ltsp ou /srv/tftp/ltsp». La gestion du serveur TFTP est assurée par inetd ou upstart. Dans le dossier TFTP de l'environnement, on trouve les différents fichiers utilisés pour le démarrage des clients. Il existe plusieurs utilitaires de boot réseau comme etherboot, gPXE, et PXE. Je n'utilise que ce dernier car la grande majorité des cartes réseau est compatible avec cette norme et les autres méthodes imposent souvent d'utiliser des supports amovibles sur les clients. Le dossier TFTP doit contenir le fichier pxelinux.0 puis le dossier pxelinux.cfg qui contient le ou les fichiers de configuration de boot. Par défaut, il n'y en a qu'un : «default». Mais il est possible de spécifier un fichier pour chaque machine. Le fichier doit avoir comme nom l'adresse IP convertie en hexadécimal (192.168.0.1 devient COA80001). Ce fichier de configuration donnera au client les noms des fichiers qu'il devra télécharger pour démarrer. Exemple :

```
DEFAULT vmlinuz ro initrd=initrd.img quiet splash nbdport=2001
```

Il est possible de personnaliser l'affichage du menu de démarrage via les outils «syslinux» et ainsi permettre à un client léger de pouvoir se connecter à différents environnements.

15 Configuration inetd

Le service «openbsd-inetd» est utilisé pour gérer les connexions ; aux images NBD, au service NBD-SWAP et au «ldm-info». Il peut être aussi paramétré pour le serveur TFTP.

```
tftp      dgram  udp    wait   root  /usr/sbin/in.tftpd /usr/sbin/in.tftpd -s /var/lib/tftpboot
9571     stream tcp   nowait nobody /usr/sbin/tcpd /usr/sbin/ldminfod
9572     stream tcp   nowait  nobody /usr/sbin/tcpd /usr/sbin/nbdswapd
2000     stream tcp           nowait nobody /usr/sbin/tcpd /usr/sbin/nbdrootd /opt/ltsp/images/base.img
2001     stream tcp           nowait nobody /usr/sbin/tcpd /usr/sbin/nbdrootd /opt/ltsp/images/public2.img
```

Par défaut, les connexions aux images NBD commencent au port TCP 2000. Il est possible de modifier le port via la commande ltsp-update-image vue plus haut.

16 LTSP-cluster

Il existe un mode «cluster» dans LTSP pour l'utilisation de clients classiques ou local-app. Pour les clients en mode autonome, ce service n'apporte rien si ce n'est la fonction de configuration des fichiers «lts.conf». Le principe est que l'on dissocie les serveurs de sessions, des serveurs d'applications grâce à un équilibrage de charge qui va répartir les clients entre les serveurs d'applications. Une interface web est utilisée pour configurer les divers paramètres. Les serveurs d'applications communiquent avec le serveur d'équilibrage pour que les clients soient répartis.

Lors de la création de l'environnement LTSP, il vous faudra utiliser l'option «--ltsp-cluster» qui vous permettra de spécifier l'adresse IP du serveur d'équilibrage.

17 Personnalisation de l'environnement

Pour personnaliser les environnements, on peut modifier les valeurs par défaut du gestionnaire de fenêtre, du fond d'écran, des applications,..., dans les dossiers /usr/lib, /usr/share, ... Il est possible aussi de définir un profil utilisateur par défaut en utilisant la fonction /etc/skel. C'est une méthode simple, rapide et indépendante des mises à jour du système.

Pour personnaliser Firefox, comme par exemple fixer le proxy web, utilisez les fonctions de lockPref intégré dans le navigateur (voir http://kb.mozillazine.org/Lock_Prefs). La même chose existe pour Chromium version libre de Google© Chrome®(voir <http://www.chromium.org/developers/how-tos/chrome-frame-getting-started/chrome-frame-administrator-s-guide>)

La création d'une borne web, dédiée à la consultation de pages sans authentification, se fait en créant un environnement «classique» puis en installant dedans un navigateur web, un gestionnaire de session et un gestionnaire de fenêtres (OpenBox est très facile à personnaliser). On crée un compte dans l'environnement LTSP qui servira à l'ouverture de session automatique (avec GDM). Pour personnaliser l'interface du navigateur web il existe des extensions (voir <http://webconverger.org/kiosk/>). Pour gérer l'accès aux sites web, le plus simple est d'utiliser un proxy qui permet de mettre à jour les acl sans modification des environnements LTSP.

Le meilleur exemple que je connaisse sur les capacités de personnalisation de LTSP est le réseau mis en place au Québec dans la ville de Laval pour les postes de leurs écoles primaires. C'est plus de 5000 clients à travers la ville qui sont connectés à un cluster LTSP.

18 Conclusion

Vous ne pourrez pas utiliser un serveur LTSP dans tous les cas de figures, comme faire du montage vidéo ou brancher des périphériques spéciaux. Pour nous, il a permis d'augmenter le nombre de postes publics avec un investissement minime. L'installation de la salle de cours de langues a coûté 900 Euro : changement seulement du poste de l'enseignant qui fait office de serveur LTSP et conversion des 15 postes datant de 2004 en clients légers. Pour les postes publics de la bibliothèque, nous avons réalisé une économie de plus de 30000 Euro en choisissant LTSP à la place d'une mise à jour de l'ancienne solution.

Le temps que je consacre à la maintenance des serveurs LTSP est de moins d'une heure par semaine (pour 4 serveurs et 150 clients) sauf dans les cas de changements ou créations d'environnement. Le plus dur n'est pas LTSP lui-même mais la configuration, des gestionnaires graphiques comme Gnome ou des applications comme la création d'un profil type pour Firefox.

19 Bibliographie et liens web

- Manuel d'administration de LTSP (anglais) ([lien direct](#))
- Projet K12Linux : <https://fedorahosted.org/k12linux/>
- Projet Kiwi-LTSP : <http://en.opensuse.org/Portal:KIWI-LTSP>
- Code source de LTSP : <https://code.launchpad.net/ltsp/>
- Page «man» de lts.conf : <http://manpages.ubuntu.com/manpages/lucid/man5/lts.conf.5.html>
- Création de plugin ltsp-build-client : <https://wiki.edubuntu.org/HowtoWriteLTSP5Plugins>
- LTSP Cluster : <https://www.ltsp-cluster.org/>