

Clients légers

Journées Réseaux 2001

10–14 décembre 2001, Lyon, France

Ce document est rédigé en commun par et grâce au travail de :

- Denis Auroux <Denis.Auroux@math.polytechnique.fr>, Centre de Mathématiques Ecole Polytechnique,
- Jean–Luc Bellon <Jean-Luc.Bellon@cpht.polytechnique.fr>, Centre de Mathématiques Ecole Polytechnique,
- Philippe Depouilly <Philippe.Depouilly@math.u-bordeaux.fr>, Laboratoire de Mathématiques Pures,
- Joël Marchand <Joel.Marchand@math.jussieu.fr>, Institut de Mathématiques de Jussieu, et
- Albert Shih <Albert.Shih@math.jussieu.fr>, Université Paris VII–Jussieu.

A) Introduction

Dans le cadre d'une rencontre à travers la liste de diffusion Mathrice : le groupe des informaticiens des laboratoires de mathématiques du CNRS (<http://www.math.cnrs.fr/mathrice>), au même moment, nous avons manifesté le besoin de déployer des solutions à base de terminaux dans un environnement Unix. Rapidement, une volonté de vouloir obtenir une solution bien adaptée et économique nous a incités à mettre nos compétences et nos moyens en commun, afin d'évaluer plusieurs solutions, partager nos expériences et en tirer une expérience.

Ce document présente dans un premier temps nos besoins en matière de postes informatiques personnels, les évolutions de ces besoins. Puis, après avoir rapidement décrit les matériels classiques, nous présenterons l'émergence des clients légers et la réponse qu'ils peuvent nous apporter.

B) Le contexte

B.2) Problématiques

Dans notre activité d'administration et déploiement des systèmes informatiques, nous avons rencontré

des problématiques sinon équivalentes du moins parallèles pour l'équipement des bureaux des chercheurs, enseignants/chercheurs et étudiants en mathématiques. L'environnement informatique de prédilection pour la bureautique et le calcul scientifiques est principalement UNIX, et les logiciels et environnements courants sont :

- une interface graphique souple, puissante et conviviale (souvent Gnome/KDE)
- des logiciels multimédia : Real-Audio, xmms (player de fichiers audio), players de fichiers video (aktion, xanim, ...)
- des logiciels de communication : netscape, courrier électronique, connexions distantes (ssh, r-commandes, ftp)
- des logiciels de traitement de texte : (La)TeX, StarOffice, AbiWord...
- des logiciels de calcul scientifique : Scilab, MuPad, Maple, MatLab, Pari, AVS ...(avec peu ou beaucoup de contraintes d'affichage graphique).

B.3) Les besoins

- Les mathématiciens ont une activité de recherche très personnelle, un travail de réflexion souvent solitaire. Dans de nombreux cas, le bruit du matériel informatique est une nuisance sonore exprimée par les utilisateurs.
- Pour le calcul scientifique, les ressources sont souvent distantes (gros serveurs de calcul), d'où une nécessité d'affichage déporté (grâce à X11). Les postes informatiques à administrer sont dans les bureaux et dans des salles blanches. Toute rationalisation de l'administration est bénéfique, car la quantité des machines à administrer augmente nettement plus rapidement que le personnel. Il est impératif de concentrer son attention sur les serveurs.
- Les étudiants accèdent dans des salles banalisées à ces serveurs. Il est impératif de garder un niveau de confidentialité et de sécurité dans ces salles, d'où la nécessité de proposer des postes à configuration fixe et limitée et si possible administrables à distance.

Le matériel répondant à ce besoin le plus souvent utilisé est le terminal X. C'est une machine qui propose un affichage déporté de systèmes UNIX, et qui possède une couche logicielle minimale : uniquement un programme, le serveur X11, sans système d'exploitation. C'est un système très verrouillé et sécurisé, mais très propriétaire. Le matériel autant que le logiciel est propriétaire d'où certains surcoûts prohibitifs lors de l'achat et des mises à jour (barrettes mémoire).

B.4) Les nouveaux besoins

- Les terminaux X sont basés sur des architectures matérielles un peu coûteuses : un coût équivalent à des PCs possédant eux des périphériques multimédia ainsi que des capacités d'affichage plus performantes. Ainsi la vitesse d'affichage d'un browser Web sur un terminal X est bien inférieure à celle d'un poste de type PC ou Mac.
- De grosses applications deviennent plus ou moins incontournables (netscape/mozilla ou visualisation OpenGL) et génèrent beaucoup de trafic réseau (pour l'affichage déporté). Ces logiciels sont ouverts dès le début de la session et restent ouverts en permanence. Ne serait-il pas plus judicieux de faire tourner ces applicatifs en local ?
- Les ressources multimédia sont de plus en plus sollicitées, les terminaux X permettant de les manipuler sont encore plus onéreux (une carte son avec une entrée coûte cher sur ces appareils), voire inexistantes.
- Enfin il peut devenir de plus en plus utile de pouvoir accéder par d'autres protocoles que X-

Window comme ICA et RDP, à des serveurs d'applications Windows (Windows TSE et Citrix MetaFrame). Ces protocoles et les clients associés ne sont pas forcément disponibles sur les terminaux X–Window actuels.

B.5) Des solutions

B.5.1) L'existant :

Les terminaux X, les stations UNIX légères ou anciennes (avec peu de puissance, utiles pour les applications légères mais permettant néanmoins les connexions distantes), les PCs Linux (équivalents aux stations légères), les PCs Windows ou les Macintosh avec un serveur X11 en tant que logiciel ajouté.

En dehors des terminaux X, les autres postes ont quatre inconvénients récurrents :

- l'administration supplémentaire
- la nuisance sonore (disque(s) dur(s), ventilateur(s))
- la sécurité :
 - disque(s) dur(s) impliquant des risques de perte de données, et donc nécessitant des sauvegardes
 - système d'exploitation complet impliquant des risques réguliers d'attaque ou d'intrusion, et donc nécessitant des mises à jour régulières
 - pour les PCs Windows et les Macintosh, système sujet aux attaques par virus
 - l'utilisateur face à une machine de type PC Windows, Macintosh, ou Unix/Linux peut s'approprier les droits d'administration de la machine (root dans le monde Unix) en un temps plus ou moins rapide, mais toujours assez faible. Aussi ces machines présentent des inconvénients structurels pour des milieux "hostiles" (salle libre–service), plus ou moins palliés par des mesures de protection partielles.
- le coût : au-delà du prix d'achat stricto sensu, les besoins annexes pour les sauvegardes, le logiciel anti–virus, et surtout les applications commerciales (même légères comme la bureautique) exécutées en local, et donc devant être achetées autant de fois que de postes.

B.5.2) De plus ou moins nouvelles solutions :

Les « e–pc » (PC sans lecteur de disquettes, plus silencieux, boîtiers réduits) : c'est bien, mais il reste souvent un disque dur (éh oui, c'est bruyant) avec des données modifiables et toujours un petit ventilateur (moins bruyant, il est vrai).

Des projets de reconversion des PC en terminaux X où on enlève le disque dur, on démarre sur une disquette avec un noyau minimal et un serveur X, voire même un démarrage via la carte réseau (avec le protocole PXE) comme le projet LTSP (<http://www.ltsp.org>) dont nous discuterons plus loin : il reste un ventilateur et du temps pour mettre en oeuvre chaque poste (du travail de reconditionnement). LTSP est un projet très proche des clients légers.

Des solutions encore propriétaires : les SunRay, terminaux totalement passifs de Sun Microsystems, nécessite un réseau rapide dédié avec un serveur SUN spécifique (avec la contrainte d'un choix d'un constructeur).

Les clients légers :

Le client léger est une appellation un peu abusive. Ni un PC, ni un terminal X. Le principe est de prendre un succédané d'architecture de PC x86 : carte mère avec bus PCI/ISA, ports PS2, parallèle,

série et USB, avec un processeur intégré (carte vidéo, processeur x86, contrôleur PCI et audio, souvent dans le même composant : processeur Géode GX1 de National Semiconductor ou un processeur à faible performance : AMK K6 266 MHz) et à faible consommation (donc sans ventilateur), le disque dur est remplacé par une mémoire flash, et le système d'exploitation réduit à sa plus simple expression (un système Linux minimal lançant un serveur X11 immédiatement au boot).

Les performances du processeur sont souvent comparables à un Pentium 233/266 MMX, ce qui peut paraître faible mais suffisant pour un serveur X11.

Au démarrage un client léger propose le même service minimum qu'un terminal X : le serveur X11 et la possibilité de faire des requêtes XDM ou de lancer des clients locaux (xterm, clients ICA ou RDP, navigateur Web local).

Ce qui a motivé le test de ces appareils :

- le besoin de proposer aux utilisateurs de terminaux X des appareils multimédia (des cartes sons plus exploitables que sur les terminaux X de base).
- des possibilités de modification de l'environnement (système Linux qui permet quelques modifications).
- la recherche d'appareils plus économiques que les terminaux X.
- la raréfaction du marché des TX : il reste un seul fournisseur, NCD, qui s'oriente de plus en plus vers les terminaux Windows purs, et ... les clients légers.

C) Évaluation de différents modèles

Nous allons aborder ici deux catégories :

- les clients légers constructeurs qui ont pour particularité d'être autonomes.
- Les clients légers « maison » qui sont en fait des PCs « diskless » (sans disque dur).

C.2) Les clients légers constructeur

Le principe :

Un noyau Linux est stocké sur une mémoire flash (disk flash IDE). Dans la plupart des cas, le flashage de la mémoire n'est pas possible car il faut un bout de code dépendant du matériel et cette manipulation est verrouillée par le constructeur. Il reste une petite zone sur la mémoire disponible pour stocker de façon permanente des petits fichiers scripts ou exécutables.

Il est possible de créer des montages NFS au démarrage, ce qui permet de mettre à disposition du système embarqué des exécutables "maison" (car compilés sous Linux). On peut ainsi utiliser des applications sur le processeur du client.

C.2.1) Les trois modèles testés

Aperçu des matériels :

Les trois modèles possèdent :

- une carte réseau 10/100 Mbits/s (le plus souvent : Raeltek 8139A)
- 2 ports série
- 1 port parallèle
- 2 ports USB
- 2 ports PS2 (clavier/souris)
- 1 port VGA

Tableau comparatif entre :

Modèle IGEL-J de IGEL (<http://www.igel.de>),

Modèle eon4000 de NEOWARE (<http://www.neoware.com>) et

Modèle Netvista 2800 de IBM (<http://www.ibm.fr>)

Modèle	IGEL-J	eon4000	Netvista 2800
Processeur	AMD K6/2 333Mhz	Geode 233 MMX	Pentium 266/MMX
Mémoire	16 Mb flash/ 64 RAM Mb	16(24) Mb flash / 64 RAM Mb	16 Mb / 64 RAM Mb
Vidéo	Embarquée S3 8Mb	Emb. 2Mb (1)	Emb. Trio S3 4Mb
Son	Compatible SB-16	Compatible SB-16	Crystal Audio 4235 (2)
ISA/PCI	1 port ISA/PCI	1 port ISA/PCI	2 ports PCI

(1) Il est possible d'ajouter une carte PCI pour une vidéo accélérée (matrox G250 par exemple).

(2) Avec haut parleur intégré.

Les deux clients légers de IGEL et de NEOWARE sont basés sur une structure de carte mère de PC dotée d'un processeur compatible Intel de performance modeste (~300 Mhz), et d'un OS stocké sur une mémoire non volatile de type <disk on chip>. Les performances en affichage déporté sous X11 sont globalement suffisantes en rapidité. Par contre la résolution liée au nombre de couleurs dans un environnement X doivent être d'au moins 1280x1024 16 plans pour pouvoir afficher des pages A4 complètes en format PDF, PS, DVI. En effet, le mode couramment utilisé depuis longtemps sur des terminaux X est de 1280x1024 et de plus la limitation à 8 plans des terminaux X pose des difficultés avec les applications et environnements graphiques de plus en plus consommateurs de couleurs, comme Netscape, KDE, StarOffice, etc... Le modèle IGEL-J répond à ces critères, alors que pour le eon4000 il faut ajouter une carte graphique dans le seul slot disponible.

Le client léger d'IBM est basé comme les autres modèles sur une carte PC standard aux performances modestes (266Mhz). Cependant contrairement aux autres modèles, il ne possède pas nécessairement de mémoire flash. Le boot se fait suivant deux techniques différentes (configuration par le BIOS):

- soit par mémoire flash avec un noyau Linux (Redhat à l'époque des tests, maintenant il s'agit de Turbo-Linux) embarquant les logiciels comme Netscape.
- soit par NFS, en vrai diskless, dans un environnement Linux.

D'un point de vue purement physique : le boîtier est 50 % plus volumineux que les IGEL, un ventilateur engendre un léger bruit qui peut être gênant pour ceux qui ont pris l'habitude de travailler sur un terminal X. Cependant ce bruit reste largement en dessous d'un PC/Macintosh habituel. Le

boîtier est relativement lourd et présente une sortie et une entrée audio permettant de brancher un micro.

L'ensemble dégage une impression de matériel bien fini et solide.

Coté logiciel :

Dans le contexte de remplacement d'un terminal X, seuls les clients légers sous Linux ont été testés. Dans les deux cas, une version de Linux installée sur le flash disk est lancée par le BIOS pour proposer une interface de configuration sur laquelle on trouve les paramètres classiques de vidéo, réseau, mise à jour de l'OS, etc...

Si la fonction première est de remplacer un parc de terminaux X, alors la configuration proposée avec le IGEL-J convient (pour le eon4000 il faut prévoir une carte graphique supplémentaire) : paramètres réseau en manuel ou par DHCP, requête XDMCP, serveur de fontes, etc...

Sur le modèle eon4000, un mot de passe système donne accès aux modifications du système de fichiers. On peut donc modifier/adapter à sa guise la version Linux installée (on a alors un accès total à la mémoire flash).

Avec le modèle IGEL-J, on est moins libre car le système de fichiers est en lecture seule. Par contre une zone est accessible (/wfs) et permet alors beaucoup d'adaptations par exemple :

- recopier dans /wfs un fichier XF86Config mieux adapté à un écran particulier,
- s'il existe, le fichier /wfs/rc.custom est un shell-script qui sera lancé dans la procédure de démarrage, on imagine alors toutes les adaptations possibles :
 - script qui modifie les configurations que l'on n'a pas pu faire avec le Setup,
 - script qui fait le montage NFS d'une partition qui contient les fontes couramment utilisées par les autres terminaux X,
 - script qui fait les montages NFS pour transformer le client léger en diskless (en mettant alors à disposition des applications locales).

Le modèle IBM fonctionne dans deux modes différents. Dans cette partie nous présentons le mode mémoire flash intégrée, le mode « diskless » sera abordé dans la partie C.3).

Dans ce cas, le fonctionnement ressemble beaucoup au modèle IGEL. Il y a une différence importante avec les deux machines précédentes. Le constructeur fournit sur son site un kit de développement de noyau ainsi que des outils nécessaires pour flasher la mémoire flash (un simple `dd if=image of=/dev/la_flash_prom`). Cela présente un avantage certain par rapport aux autres solutions, car il est très facile d'installer un nouveau noyau si par exemple nous voulions essayer les ports USB (non supporté dans Linux 2.2.x (x < 18)).

Mais comme tout avantage est en général accompagné d'un désavantage, l'image ainsi créée est très volumineuse et il ne nous a pas été possible d'inclure, dans les 48 Mo du modèle que nous avons, autre chose que le strict minimum à savoir un noyau, le serveur X11 et fvwm2. Nous n'avons donc pas pu avoir Netscape en mémoire Flash. Il faudrait savoir ce qu'il est possible de faire avec 96 Mo mais vu le prix des mémoires flashes....

L'utilisation du Kit ne pose pas de problème majeur. Notre serveur NFS était sous FreeBSD et cela n'a

posé aucun problème de compatibilité.

Ce type de solution présente certainement des avantages (codes ouverts du système de patch et de flash). Malheureusement ils sont très largement occultés par le prix prohibitif du matériel : 900 \$ (sur le site US de IBM). C'est-à-dire plus cher qu'un PC classique pouvant aussi être transformé en client léger grâce à une carte réseau supportant le protocole de boot PXE et proposant alors une autre vitesse d'horloge (cf C.3).

Amélioration du fonctionnement des clients légers :

Si maintenant on veut faire plus de choses qu'avec un terminal X, les clients légers proposent par exemple des clients locaux, window manager (fvwm95), butineur (Netscape) et des périphériques de stockage (lecteur de disquettes, CDRom) et 2 ports USB. Que ce soit pour un client local comme Netscape ou pour un périphérique de stockage (USB ou non), le problème du partage de fichiers se pose et la seule solution NFS – qui ne propose pas d'authentification entre l'utilisateur et son serveur – ne convient ni pour les bookmarks de Netscape, ni pour le transfert de fichiers entre le répertoire de travail de l'utilisateur et le lecteur de disquettes local. Pour ces problèmes de visibilité des fichiers personnels, beaucoup de clients locaux sont difficilement utilisables et des solutions sont à expérimenter sur des configurations en mode diskless avec SAMBA (smbfs et les commandes smbmount/smbumount) par exemple. Des informations sont disponibles sur le développement d'applications autour du eon4000 sur <http://www.neolinux.org>.

Par contre, pour le déport du son depuis l'application lancée sur le serveur vers la carte audio du client léger, voici 2 solutions :

La première approche consiste à installer sur le client léger les applications telles que RealPlayer, Xmms, ..., en les installant sur une partition du serveur qui sera montée par NFS. Simultanément, il convient d'installer sur le serveur des scripts se chargeant de lancer le programme sur le client, via par exemple rsh ou même ssh (si ce dernier a été également installé sur le client).

Toutefois, RealPlayer est fréquemment utilisé en tant que plug-in d'un navigateur web, et au contraire de la version "stand-alone", il semble impossible de configurer le plug-in pour une exécution distante sur le client léger, alors que le navigateur s'exécute sur le serveur. Même s'il est probablement possible, avec les compétences suffisantes, d'écrire un plug-in de substitution qui passerait par le réseau pour appeler une version s'exécutant sur le client léger, il devient pratiquement obligatoire d'installer également le navigateur web sur le client léger. Ceci pose néanmoins problème, car il est probablement irréaliste d'installer le module de messagerie et tous les autres appendices du navigateur sur le client léger, à moins de le transformer intégralement en station diskless. Il devient également nécessaire de mettre en place une procédure d'authentification des accès aux applications locales du client léger.

La seconde approche est par exemple celle que propose VirtualFS/RemAudio (<http://www.solucorp.qc.ca/virtualfs/>). Il s'agit d'installer d'une part sur le serveur une version modifiée de la librairie standard du système, et d'autre part sur le client léger un module "serveur". Lors de l'exécution d'applications sur le serveur, la librairie modifiée intercepte les appels systèmes tels que open(), read() et write() portant sur certains périphériques, et les transmet via le réseau au module installé sur le client léger, lequel effectue les requêtes correspondantes sur les périphériques locaux.

Nous avons toutefois expérimenté des difficultés d'installation de VirtualFS, car celui-ci exige une infrastructure délicate à mettre en oeuvre, notamment des versions patchées des librairies standard

du système Linux, qui ne fonctionnent qu'avec certaines distributions. En outre les performances en lecture audio sont assez limitées et certaines applications ne fonctionnent pas tandis que d'autres produisent un son fortement haché.

Ceci a conduit au développement d'un outil équivalent à VirtualFS mais se restreignant à la fonctionnalité sonore uniquement, ce qui lui permet d'être beaucoup plus léger et plus performant grâce à un code réseau optimisé.

Ce logiciel (disponible sur : <http://www.math.polytechnique.fr/cmat/auroux/prog/iaudio-0.1.tar.gz>) se compose lui aussi de deux modules :

- le module serveur (iaudioserv), qui devra être installé sur le client léger et exécuté lors du démarrage. Sa petite taille (8 Ko une fois strippé) permet de le logger directement dans le répertoire /wfs du client IGEL, ce qui évite d'avoir recours aux disques du serveur ; il convient simplement d'ajouter une ligne de la forme "/path/iaudioserv &" vers la fin du script rc.custom.
- d'autre part, un binaire objet partagé (iaudio.so.0.1.0), également de petite taille (13 Ko), qui restera sur le serveur; ce module fournit un remplacement aux fonctions open(), read(), write(), etc... de la librairie standard. L'insertion s'effectue sans modification du système et sans droits particuliers (sauf pour les programmes suid), en ajoutant parmi les variables d'environnement la valeur "LD_PRELOAD=/path/iaudio.so.0.1.0"; cette variable indique à l'éditeur de liens dynamiques (ld.so) d'insérer le module iaudio.so en supplément aux librairies standard lors du lancement de toute application. Cette modification de la valeur de LD_PRELOAD peut être effectuée soit dans les fichiers de configuration de l'utilisateur pour affecter toutes les applications, soit dans des scripts encapsulant les diverses applications pouvant être amenées à produire des sons.

Lorsque l'application accède aux périphériques audio, le module iaudio.so intercepte les appels système et les transmet au programme iaudioserv en train de tourner sur le client léger, lequel effectuera les opérations correspondantes sur les périphériques audio du client léger. La redirection s'effectue par défaut vers la machine dont le nom figure dans la variable d'environnement DISPLAY; il est possible de modifier ce choix à l'aide de la variable d'environnement NETAUDIOHOST. Les performances permettent de faire passer sans problème du son de qualité CD sur un réseau local 10 Mbps ; toutefois il faut rappeler que la bande passante occupée par un tel signal est nécessairement de l'ordre de 200 Ko/s, ce qui peut poser des problèmes d'encombrement. Par ailleurs, si une telle approche de bas niveau fonctionne très bien lorsque le client et le serveur ont des systèmes d'exploitation relativement proches (par exemple Linux de chaque côté), elle est moins efficace lorsque les gestions des périphériques, notamment les ioctl() associés, diffèrent trop.

C.3) Les clients légers « diskless »

C.3.1) Client léger IBM en mode diskless :

En utilisant un autre kit de développement fourni par IBM sur leur site, il est possible d'avoir les patches d'un noyau Linux pour qu'il soit supporté par la NetStation. Une fois qu'on a constitué un noyau et les librairies nécessaires, on exporte depuis un serveur NFS l'arborescence Linux. Depuis le BIOS de la Netstation, on lui indique le numéro IP du serveur NFS et le chemin d'accès du noyau Linux. Nous obtenons alors un poste de travail où la plupart des opérations peuvent être effectuées sur

le processeur en local (Netscape, etc...).

C.3.2) Les clients LTSP (<http://www.ltsp.org>) :

Les clients LTSP s'apparentent aux clients légers dans la démarche : utiliser des architectures matérielles bien connues (à base de PC) avec un OS libre donc adaptables à nos besoins. L'idée du projet LTSP est assez simple : proposer une suite logicielle minimale (un système Linux simplifié, un serveur X11 et des applications Internet) se téléchargeant au démarrage du PC via un serveur NFS.

Ce que nous avons mis en place et testé est la configuration suivante :

Côté matériel :

- carte mère i810 (maintenant i815E) intégrant :
- Intel Celeron 700 Mhz
- 64 Mo de RAM
- carte réseau Intel EEPRO100 (10/100 Mb/s – support PXE)
- carte son compatible Sound Blaster
- carte vidéo AGP avec accélération 3D jusqu'à 11 Mo de RAM (exploitée avec Xfree86-4.1)
- 2 ports USB
- 2 ports série, un port parallèle
- 2 connecteurs IDE (un de libre)
- 1 lecteur CDROM
- 1 lecteur de disquettes
- mini boîtier (par exemple : dimensions : 300x320x78 mm) équipé d'un mini-ventilateur

Le matériel testé provenant de trois constructeurs différents (voir les références) est de bonne finition et très standard. Les performances sont significativement supérieures aux clients légers constructeurs ; la carte vidéo 3D de la carte mère Intel est très bien gérée par Xfree86-4.1. On peut ainsi obtenir de très bonnes performances pour des affichages de type OpenGL déportés (notons que NEOWARE le propose aussi par ajout de carte vidéo 3D sur le slot PCI avec les drivers associés).

Côté logiciel :

Le client LTSP ne contient aucun support pour du logiciel en dehors du protocole PXE présent sur la carte réseau. Le protocole PXE permet au client de réclamer une adresse IP via le protocole DHCP et ensuite de télécharger sur un serveur offrant un service PXE n'importe quel logiciel, en général un logiciel de démarrage (boot-loader). Le boot-loader permet le téléchargement, via tftp par exemple, d'un système d'exploitation.

Dans notre cas, nous nous sommes rapprochés du principe des clients légers classiques : le système d'exploitation est un noyau Linux simplifié. Comme nous n'avons aucun support matériel pour archiver du logiciel (disque dur ou flash-disk), nous utilisons une fonctionnalité du système Linux qui s'appelle NFS-ROOT (<http://linux.uhp-nancy.fr/linuxdoc/HOWTO/mini/NFS-Root.html>) : le noyau se télécharge via tftp et l'ensemble du système de fichiers réside sur un serveur NFS (en l'occurrence le même que le serveur DHCP et PXE). Le serveur possède dans un répertoire particulier (classiquement /tftpboot) une arborescence complète du système Linux à déployer.

Il va sans dire que dans ce cas la mise en oeuvre des fonctionnalités souhaitées n'est pas limitée ni par la mémoire flash, ni par des contraintes constructeurs. Les versions des systèmes, des applications ou des intégrations dans un environnement particulier sont totalement à la charge de l'administrateur. Sur le serveur, la racine du système des clients est unique, les scripts qui sont propres à chaque client prennent en fait comme argument une variable d'environnement propageant l'adresse IP du client obtenue via DHCP.

Remarque : ceci est fort similaire du fonctionnement de machine IBM en mode diskless, notamment la séquence de boot. La différence est que le procédé PXE est remplacé sur l'IBM par l'intégrateur du boot-loader avec un setup ad-hoc, dans le BIOS de la machine.

Le projet LTSP propose via des packages précompilés une mise en oeuvre simple et assistée des clients. Il ne faut quand même pas sous-estimer le travail de mise en route. Dans notre cas, nous avons mis une bonne journée pour mettre en oeuvre un noyau efficace avec l'accélération 3D/OpenGL (personne connaissant Linux et son administration). Les documentations étant assez nombreuses (et francisées). Ensuite, le déploiement ne nécessite aucun travail supplémentaire, car les clients, totalement dépourvus d'autonomie, n'ont besoin d'aucune configuration, uniquement le déballage et le branchement (en dehors bien-sûr de règles de base en matière de sécurité : blocage dans le BIOS par un mot de passe du boot via disquette et CDRom).

Le projet LTSP est suffisamment ouvert pour permettre facilement l'installation d'un noyau recompilé pour un ajout de pilotes ou de fonctionnalités supplémentaires. Dans un environnement Linux, cette solution est très efficace.

La mise en oeuvre de cette solution demande donc un travail d'administration préalable obligatoire, des connaissances minimales dans le système Linux. La liberté d'action est obligatoirement associée à un investissement préalable. Il faut tout de même modérer cet investissement en comparant le travail nécessaire pour reconfigurer, adapter les clients légers classiques. Un autre inconvénient est la persistance d'un ventilateur (même petit) qui produit une légère nuisance sonore (pour les adeptes du silence total). A ce propos, il serait souhaitable de trouver de tels appareils avec des alimentations externes par exemple.

Le client LTSP peut ensuite s'utiliser comme poste local avec toutes les fonctionnalités d'un poste classique ou en tant que pur client X11 (mais avec des aménagements comparables aux autres clients légers : déport de l'accès des périphériques son et magnétiques par exemple).

On obtient donc une architecture qui rappellera bien des souvenirs aux personnes ayant connues il y a plus de 10 ans les stations Unix sans disque (Sun 3, Digital, H-P). Il faut souligner que la disparition de ces machines avaient été motivées en partie par la faiblesse relative des serveurs NFS et des réseaux locaux (Ethernet 10 Mbs partagés) et le coup important de la mémoire centrale qui ne permettait pas d'éviter le swap par le réseau. Aujourd'hui, un réseau local 100 Mbs commuté permet de faire de tels montages d'arborescence rapidement et sans souci pour les voisins. De plus la mémoire peut être aisément dimensionnée à une valeur suffisante pour que la session de l'utilisateur tienne en mémoire. Enfin les serveurs NFS peuvent désormais tenir une charge nettement suffisante pour supporter essentiellement la phase de démarrage de tels environnements. Aussi ne faut-il pas rejeter pour les raisons du passé ces configurations. Reste à évaluer quels bénéfices on donne aux utilisateurs à faire tourner leurs sessions en local sur leur poste personnel, plutôt qu'en mode partagé sur un gros serveur multi-processeurs avec plein de mémoire. Cela dépend en bonne partie des outils manipulés, et aussi de la philosophie des personnes : attachement ou pas à la notion de poste/station personnelle.

D) Conclusions

Les clients légers sont une alternative aux terminaux X, d'un coût plus faible, avec plus de possibilités d'adaptation aux besoins de chaque structure et promis à succéder de toutes façons à ces terminaux X qui vont disparaître du marché. Ce type de solution ont aussi l'avantage de ne pas se démoder trop rapidement, contrairement aux PC/Mac, dont la durée de vie est de l'ordre de 3 à 4 ans, si on veut profiter des dernières mises à jour système et surtout logicielles. Pour donner un ordre de comparaison, bon nombre de terminaux X, installés il y a plus de 6 ans (voire 8 ans pour les plus anciens), continuent de fonctionner normalement. Avec les mises à jour régulières des serveurs, les utilisateurs profitent, sans changer leurs postes de travail, des dernières applications et dernières fonctionnalités système. Sur une telle durée de vie, le gain financier et de temps humain est alors très important.

Il reste néanmoins des ombres sur ces clients légers :

- le matériel manque quelques fois de qualité : assemblage très mécanique moyen, choix de composants sans performances (carte vidéo rudimentaire par exemple)
- l'absence de ventilateur permet de ne pas avoir de bruit, mais induit alors une forte chaleur (le boîtier devient vite brûlant). Quel impact cela aura-t-il sur ces matériels, comparativement aux terminaux X "inusables" ?
- chez IGEL, le suivi logiciel n'est pas toujours suffisamment approfondi, leurs clients ne savent toujours pas utiliser les ports USB, car le noyau en est encore à la version 2.0.36, les mises à jour sont assez lentes
- les outils de déploiement de clients légers logiciels sont assez imparfaits, bien que NEOWARE s'en sorte assez bien en terme de support logiciel et amélioration des fonctionnalités avec leur logiciel ezRemote-Manager
- les coûts des clients légers diffèrent assez sensiblement pour des performances assez comparables, l'environnement logiciel devenant alors un critère plus important que la nature du matériel.

La solution LTSP est assez intéressante, mais ne convient pas à des déploiements sans serveur de logiciels et nécessite un certain travail préparatoire. Les constructeurs ou assembleurs n'offrent pas à notre connaissance de postes totalement silencieux de ce type.

Et Windows dans tout ça ?

Il est tout à fait possible d'utiliser des clients ICA, soit locaux, soit sur le serveur via X11, qui permettent l'utilisation dans un contexte très robuste d'applications Windows. Ceci permet donc de considérer ces clients légers comme des postes vraiment complets, et permettant d'accéder dans de bonnes conditions aux deux mondes Unix et Windows.

Il est aussi possible de mettre en oeuvre des solutions du type Vmware, Win4Lin, VNC, plex86 (en développement) ou Wine (idem). Ces dernières solutions restent quand même lourdes à déployer et administrer sur un réseau de grande taille.

E) Références

Les différents constructeurs testés :

- IGEL J et W : <http://www.igel.de>

- Eon 4000i et 4000s : <http://www.neoware.com>
- IBM NetVista 2800 : <http://www.ibm.fr>
- Flat-Desk 820 – Transtec : <http://www.transtec.fr>
- NetSentia Book-PC – Elonex : <http://www.elonex.fr>
- E-Vectra – Hewlett-Packard : <http://www.france.hp.com>

Les supports logiciels pour :

- **IGEL** : <ftp://ftp.igel.de/pub/firmware/>
- **NEOWARE** : <http://www.neoware.com/support.html>, <http://www.neoware.com/beta> et <http://www.neoware.org>
- **IBM** : <http://service.boulder.ibm.com/nc/linux/linux.shtml>
- **Clients LTSP** : <http://www.ltsp.org>,
<http://linux.uhp-nancy.fr/linuxdoc/HOWTO/Diskless-HOWTO.html>,
<http://linux.uhp-nancy.fr/linuxdoc/HOWTO/Diskless-root-NFS-HOWTO.html>,
<http://linux.uhp-nancy.fr/linuxdoc/HOWTO/Network-boot-HOWTO/index.html>,
<http://linux.uhp-nancy.fr/linuxdoc/HOWTO/mini/NFS-Root.html>,
<http://linux.uhp-nancy.fr/linuxdoc/HOWTO/mini/NFS-Root-Client-mini-HOWTO/index.html>,
avec les versions françaises sur : <http://linux.uhp-nancy.fr/HOWTOFRENCH/INDEX.html>

Améliorations des fonctionnalités (déport du son) :

<http://www.solucorp.qc.ca/virtualfs>,
<http://www.math.polytechnique.fr/cmat/auroux/prog/iaudio-0.1.tar.gz>