MacOS X en environnement réseau

Laurent Blain

MacOS X en environnement réseau

Laurent BLAIN
LAAS-CNRS
7 Avenue du Colonel Roche 31077 Toulouse Cedex 4
Laurent.Blain@laas.fr

Résumé

MacOS X est une évolution majeure du système d'exploitation d'Apple présentant une interface utilisateur évoluée et basé sur un noyau UNIX. Ce changement nous a poussé à nous y intéresser pour savoir s'il était un choix possible comme poste de travail individuel, à la fois pour les aspects bureautiques mais aussi le développement UNIX. Pour cela, il fallait qu'il s'intègre correctement dans le réseau du LAAS, sous la même forme que les postes UNIX et Windows existants. Nous avons donc étudié les possibilités de connexion à un annuaire et choisi LDAP et les différents schémas nécessaires. Nous avons ensuite étudié le fonctionnement des différents automonteurs en particulier pour les répertoires utilisateurs. Les protocoles NFS et CIFS comme moyen d'accès aux répertoires partagés ont été comparés, et nous avons montré que NFS était le seul réellement utilisable actuellement. Nous avons ensuite étudié d'autres aspects comme l'impression avec CUPS et l'administration distante et automatisée de MacOS X. Nous avons donc montré les possibilités, mais aussi les limites de cet OS.

Mots-clefs

MacOS X, LDAP, Automontage

1 Introduction

Cet article a pour but principal de montrer comment nous avons commencé et souhaitons déployer MacOS X dans l'environnement réseau du LAAS. Nous présentons tout d'abord brièvement un historique du poste de travail au laboratoire, et en quoi MacOS X, un nouvel UNIX, s'inscrit dans cette évolution, et ensuite la configuration actuelle du poste de travail au sein du réseau du LAAS. Dans un deuxième temps, nous présentons les spécificités de MacOS X et comment il a fallu procéder pour mettre en œuvre l'intégration de cet OS dans l'architecture existante. Nous avons mis l'accent sur l'étude des annuaires, plus particulièrement LDAP, et du montage de répertoires, en donnant cependant quelques éléments sur les autres aspects comme l'administration distante et l'impression. Nous terminons en montrant les quelques limites que nous avons trouvées et les nouveaux services que nous souhaitons mettre en œuvre dans le futur.

2 Le poste de travail au LAAS: micro-ordinateurs et stations de travail

Dans un premier temps, nous allons décrire l'histoire du poste de travail individuel au LAAS, et dire que peut être MacOS X est une solution à la divergence entre stations de travail et postes bureautiques existant depuis le début.

2.1L'histoire

Le LAAS a , pour le "choix" de ses ordinateurs, une histoire qui est très liée aux deux grand types d'utilisation existantes :
• la bureautique,
• le développement.
En raccourci on peut dire que les utilisateurs ne faisant que de la bureautique se sont tournés vers le monde micro (Mac ou PC sous Windows), alors que pour les développeurs, le choix a été celui de la station de travail Sun. L'histoire récente a fait évoluer les choses, et remet en cause nos choix et modes de fonctionnement, en particulier avec une convergence de ces deux activités vers un seul type de machine.
Qu'est ce qui caractérisait un micro-ordinateur individuel :
• un OS mono-utilisateur simple et peu fiable,
• une interface utilisateur très "user friendly",
• de nombreux outils et applications,
• Microsoft Office,
• une administration cauchemardesque.

Qu'est ce qui caractérisait une station de travail :
• un OS fiable (UNIX) et multi-utilisateurs,
• une interface utilisateur complexe,
• un nombre d'applications limité, en particulier pas Microsoft Office,

• des outils de développement,
• une administration locale et en réseau aisée.
Ces deux types de configuration étaient insatisfaisantes. Les micros ne pouvant être administrés correctement l'équipe système ne pouvait s'en occuper comme il aurait fallu sauf à passer un temps très important sur ces machines. De plus la fiabilité très faible des OS entraînait de gros problèmes. Cependant, les utilisateurs de ce type de poste n'auraient pas souhaité passer à un autre type de poste de travail dont l'interface est beaucoup moins aisée à manipuler, et le nombre d'applications plus réduit. D'autre part, les utilisateurs de station de travail se retrouvaient avec un gros problème pour échanger des documents dans le format le plus répandu, c'est-à-dire Microsoft Word. D'ailleurs depuis qu'au laboratoire nous avons mis en place des serveurs Windows multi-utilisateurs accessibles à partir de stations UNIX, de nombreux utilisateurs s'y connectent en permanence pour travailler sous Office.
L'idéal serait donc une convergence de ces deux systèmes :
• la fiabilité et la facilité d'administration réseau des stations de travail,
• la simplicité de l'interface et la richesse d'application des micros.
Si on regarde l'évolution des systèmes, on voit que ces deux modèles convergent, mais peut être plus rapidement dans un sens que dans l'autre.

	Micros/Bureautique/Autogérés	***Station de travail/ Développement/Administrés en réseau***
Avant-hier	PC/Windows Mac/MacOS 6,7,8,9	Sun/Solaris
Hier	*Arrivée de* PC/ Windows NT4/2000/XP **administrés**	*Arrivée de* PC/Linux
Aujourd'hui	*Arrivée de* Mac/MacOS X **administrés**	
Demain	PC/Windows XP(Windows-Microsoft Office) PC/Linux (Gnome ou KDE-OpenOffice) Sun/Solaris (Gnome-StarOffice) Mac/MacOS X (Aqua-Microsoft Office)	

2.2Les micro-ordinateurs individuels autonomes

Ce type de machine disparaît et c'est tant mieux pour nous les administrateurs systèmes. D'un autre côté, cela veut dire que nous aurons encore plus de machines à gérer. Le dernier Windows personnel est Windows Me, et à partir de Windows XP on peut considérer que tous les PC Windows ont un système multi-utilisateurs, fiable et administrable. Apple a franchi le pas de son côté avec MacOS X. Le problème ne sera donc plus de savoir si on peut administrer ces machines, mais comment on les administre. Le problème des PC s'est posé il y a 4 ans au laboratoire et une solution a été trouvée. Il faut maintenant se pencher sur les Mac.

2.3Les stations de travail de développement

Un autre problème est le choix d'une station de travail. Sun était la solution choisie jusqu'alors pour différentes raisons :
• un OS réseau fiable de type UNIX,
• des applications (en particulier en micro-électronique),
• un matériel maintenable en volume,
• l'adhésion aux grands standards.
Un premier point important est le fait que l'interface utilisateur de SUN n'est pas encore aussi élaborée que celle que l'on peut trouver sous Windows ou MacOS X (Aqua). L'interface utilisateur ce n'est pas uniquement l'aspect des fenêtres et la gestion de la souris, mais aussi la communication entre applications, l'accès aux périphériques à partir de toutes les applications. De ce point de vue-là, SUN est encore à la traîne. Les applications n'ont pas suivi les évolutions.
D'autres éléments entrent aussi en compte. SUN vendeur de stations de travail est maintenant avant tout un vendeur de serveurs et d'outils de développement Java. Est ce que les stations vont évoluer ? Qu'en est-il de Java Desktop System, ce projet de système de bureau basé sur Linux (SUSE) et donc une plateforme Intel ? Toutes ces questions nous poussent à regarder ce qui se fait aussi ailleurs.
Il y a d'abord et surtout les PC à architecture Intel. Sur cette plateforme, Windows XP n'est pas une bonne solution pour nous car les développements sur cet OS restent très limités au LAAS. Mais, il y a maintenant d'autres OS UNIX : FreeBSD, OpenBSD, et surtout Linux qui offre toutes les caractéristiques nécessaires à la station de travail : OS fiable, administrable, interface graphique élaborée, applications de plus en plus nombreuses, outils de développement connus. Linux pourrait être la solution s'il était possible de choisir une distribution et une interface graphique stables, et du matériel maintenable et supporté de manière à pouvoir le déployer en volume et de s'y tenir un certain temps. En effet, le nombre de distributions est relativement important, il y a au moins 2 interfaces graphiques concurrentes et aucune n'est au même niveau de cohérence et de complétude que celles des solutions propriétaires.

2.4Bureautique et Développement : une synthèse ?

La question qui se pose donc actuellement est : est-il possible d'intégrer ces deux fonctionnalités sur une même plate-forme et est ce souhaitable ? A priori oui, car les utilisateurs de stations SUN sont de plus en plus nombreux à demander des

outils de bureautique plus ou moins sophistiqués : traitement de texte, outil de présentation, tableur, outil de dessin, outil de manipulation graphique, navigateur, lecteur multimédia, outil de mail, génération de PDF, synchronisation PDA...
Peu d'OS supportent toutes ces applications : MacOS X peut être une solution à côté de Windows et Linux. Comme nous l'avons dit (cf. 2.3), Windows même en version XP, ne peut pas être adoptée comme station de développement. Cet OS est trop fermé, et notre culture du développement trop basée sur UNIX pour envisager de changer. Ce système restera donc cantonné à la bureautique et à quelques développements spécifiques. La solution Linux est insatisfaisante pour les raisons indiquées ci-dessus (cf. 2.3), même si nous ne l'abandonnons pas. Nous nous penchons donc sur la solution MacOS X. A priori celui-ci apporte tous les éléments dont nous avons besoin :
• une interface utilisateur complète, simple et reconnue (Aqua), profitant au mieux de l'OS sous-jacent,
• de nombreuses applications, commerciales ou libres, dont Microsoft Office, du monde Mac ou du monde UNIX/X11,
• un support de très nombreux périphériques,
• un OS de type UNIX fiable et administrable en réseau,
• le support des outils de développement connus (GNU),
• une variété matérielle limitée.
Donc MacOS X est composé à la fois d'un système d'exploitation UNIX et des deux types d'interface permettant de l'utiliser et de le gérer, le mode terminal et un système de fenêtrage, tous les deux très complets. Est ce la station UNIX que nous attendions ? Nous avons commencé à étudier l'administration de cet OS dans la configuration habituellement mise en place au LAAS.

3 La configuration réseau

Notre architecture réseau est relativement simple. Tout poste de travail, quel que soit son OS, est indifférencié et doit permettre à n'importe quel utilisateur d'accéder à la fois à ses fichiers (son répertoire utilisateur), ses applications, ses imprimantes. Un poste de travail ne contient aucun fichier nécessitant une sauvegarde.
Cela signifie que les données sont sur des serveurs, serveurs de fichiers, d'application ou d'impression, et que ceux-ci sont eux sauvegardés. Cela implique aussi la mise en place d'un ou plusieurs annuaires synchronisés permettant d'identifier les utilisateurs et d'accéder à ces ressources. Ce modèle est donc assez éloigné du modèle micro traditionnel où tout est installé localement, applications et données. Nous avons cependant déjà déployé ce type d'architecture sur les PC Windows avec succès. Cela doit donc être possible sur les Mac.

3.1Un poste connecté au réseau

Tout poste de travail est obligatoirement connecté au réseau puisque les ressources sont toujours sur des serveurs distants. Cela ne pose plus de problèmes aujourd'hui étant donné que dans chaque bureau ou salle de manipulation, il existe maintenant des prises réseau permettant l'accès immédiat au réseau local.
Ce qui est nouveau, ce sont maintenant des utilisateurs de portables qui veulent accéder à leurs données de chez eux ou sur un autre site. Sous Windows, la solution est d'utiliser les fichiers hors-connexion qui permettent de maintenir un cache local de dossiers et fichiers avec une synchronisation lors de la reconnexion au réseau. Pour d'autres OS tels Linux ou MacOS X, il faudra trouver une solution du même ordre.

3.2Un compte unique

Un compte est caractérisé par un nom de login, un mot de passe et un répertoire utilisateur. Ceux-ci doivent être uniques. Pour le nom de login et le mot de passe, cela implique que les différents annuaires utilisés au laboratoire permettant de se connecter aux machines soient synchronisés : NIS+, Active Directory et LDAP. Il "suffit" de mettre en places des outils de gestion de comptes (création, modification, suppression) qui interviennent sur ces différents annuaires. L'utilisateur change aussi son mot de passe au travers d'un outil spécifique qui propagera les modifications vers les différents annuaires.
Les répertoires utilisateurs sont sur des serveurs Solaris : l'accès à ceux-ci se faisant soit via NFS pour les postes UNIX, soit via CIFS (Samba) pour les postes Windows. Un même répertoire est vu comme `/home/user` (UNIX) ou comme `H:` (Windows).

3.3Des serveurs robustes

Les ressources sont sur des serveurs fiables et sauvegardés. Ce sont des serveurs sous Solaris et Windows 2000. Il y aura sûrement un serveur sous MacOS X Serveur pour les Mac.
La répartition des services est la suivante en fonction du client :

	Client UNIX	***Client Windows***
Authentification	Serveur NIS+ Solaris	Serveur Active Directory Windows
Fichiers	Serveur NFS Solaris	Serveur CIFS Solaris
Applications	Serveur NFS Solaris	Serveur CIFS Windows
Imprimantes	Serveur LPD Solaris	Serveur CIFS Windows

3.4L'administration du poste

Un autre point important est la volonté de ne pas donner des droits d'administration aux utilisateurs des machines. C'est le cas depuis toujours sur les stations Sun. Nous l'avons aussi mis en place sur les PC Windows avec certaines difficultés pour des applications qui nécessitent des droits supérieurs aux droits utilisateurs. Il faut espérer que les applications MacOS X seront correctement écrites et qu'elles s'exécuteront sans problème pour un utilisateur standard.
Cette restriction des droits prévient toute modification du système et toute installation de nouvelle application ou matériel entraînant aussi une modification du système. Il faut bien sûr traiter à part le cas des portables où l'utilisateur peut avoir sur un autre site à installer un matériel ou une application. Dans ce cas, nous créons un compte local à la machine supplémentaire qui dispose des droits suffisants.
Cette administration non-déléguée implique des moyens d'administration à distance pour pouvoir intervenir sur les postes le plus facilement possible, et automatisés pour gérer l'ensemble des ressources partagées (comptes, fichiers, imprimantes...). Pour les stations UNIX et les ressources communes, cela est réalisé par des scripts (Shell ou Perl), et pour les PC par une utilisation des stratégies Active Directory et par un outil de contrôle à distance. Enfin toute machine Solaris ou Windows 2000/XP peut être installée automatiquement par le réseau.

4 Gestion des Mac

MacOS 6, 7, 8 et 9 étaient des systèmes d'exploitation dédiés au poste de travail individuel autonome, même s'ils intégraient déjà quelques fonctions réseaux pour l'accès aux imprimantes et le partage de fichiers. Mais clairement ils n'avaient pas la vocation à être administrés, ni n'avaient la fiabilité qu'on pouvait espérer d'OS modernes. Même Windows avait franchi le pas avec NT. Donc jusque-là le support technique était limité à quelques interventions pour résoudre des problèmes critiques : mais aucune politique d'administration ne pouvait être mise en place. Ceci pouvait être gênant car les Mac étaient quand même répandus dans 2 groupes de recherche, et surtout dans les services administratifs et les secrétariats.
L'arrivée de MacOS X nous a poussé à nous intéresser à ce système pour 2 raisons :

- comme indiqué ci-dessus, l'équipe administration système souhaitait assurer le même service de support (dépannage et aide à l'utilisation, sauvegarde des données, mise à disposition de ressources réseaux (applications, imprimante , mail)) aux utilisateurs de Mac,
- d'autre part, MacOS X nous paraissait intéressant aussi comme une nouvelle plate-forme de développement en complément ou remplacement des stations Sun.

Donc, dans un premier temps nous avons étudié ce système pour voir comment nous pourrions l'intégrer dans l'architecture existante du laboratoire, en conservant si possible le modèle de poste de travail décrit ci avant. Dans les chapitres suivants, nous présenterons les résultats de cette étude en sachant bien que MacOS X est un système jeune qui risque de connaître encore des modifications majeures dans les prochaines versions.

4.1MacOS 9

MacOS 9 était et est toujours un système mono-utilisateur, mono-tâche, avec de faibles ressources réseaux (pas de client NFS, CIFS, pas d'intégration dans un annuaire) donc impossible à administrer correctement. Le symptôme de la faible fiabilité de MacOS 9, la bombe, est celui-ci :

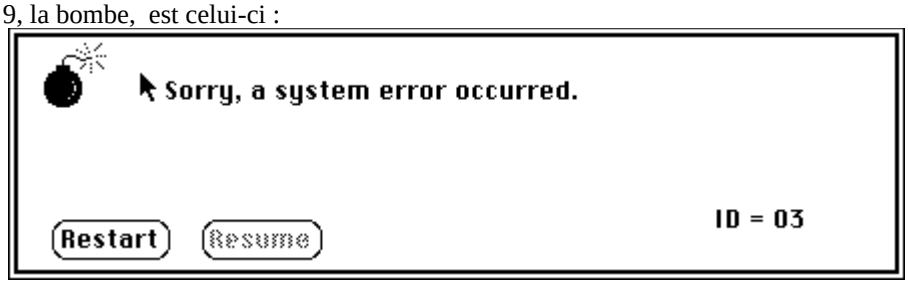

Figure 1 – *La bombe MacOS 9*

4.2MacOS X = UNIX

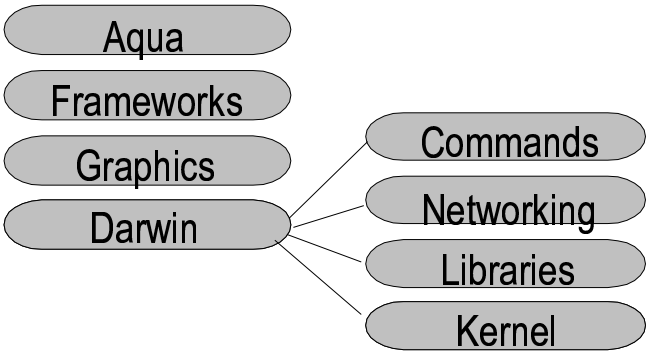

Figure 2 – *La structure de MacOS X (extrait de [1])*

On peut dire que Darwin, base de MacOS X, est un OS UNIX basé sur un micro-noyau Mach plus un ensemble de composants noyaux BSD, entrées/sorties, réseau et systèmes de fichier, de bibliothèques et de commandes (Figure 2). Il est manipulable et utilisable comme un système UNIX que ce soit du point de vue des commandes, de son fonctionnement, de sa configuration ou du développement. L'interface graphique propriétaire est Quartz, mais Apple propose déjà un serveur X11 qui sera complètement intégré à la prochaine version (Panther). Donc, même les applications UNIX/X11 pourront s'exécuter ou s'afficher sur cet OS.
Ce sont ces aspects UNIX qui nous intéressent en particulier du côté administration puisque à priori cela signifie des modes de fonctionnement proches de ceux de Solaris. Nous devrions donc pouvoir mettre en place pour MacOS X les services existants sur les machines Sun (annuaire unique des comptes, partage de fichiers...), en réutilisant les outils déjà développés (scripts...).

5 Service d'annuaire

Dans un environnement réseau, il est absolument nécessaire d'utiliser un annuaire pour gérer de manière centralisée au moins les comptes utilisateurs, mais aussi les montages de systèmes de fichiers, les imprimantes… Le LAAS a adopté NIS depuis le choix de stations Sun comme stations de travail. Depuis nous sommes passés à NIS+ avec une compatibilité NIS pour les systèmes UNIX autres que Solaris (HP-UX, BSD, Linux). Depuis peu, nous avons commencé à mettre en place un annuaire LDAP, qui devrait à terme remplacer NIS+, pour l'authentification sur d'autres types de systèmes ou d'applications tels les serveurs Web. Enfin, nous disposons aussi d'un annuaire Active Directory. Tous ces annuaires sont synchronisés.
Le format "standard " d'annuaire de MacOS X est NetInfo. Ce format étant assez peu répandu, nous n'avons pas souhaité l'utiliser. Le futur semble nous donner raison puisque ce format devrait disparaître ou tout au moins ne plus être le format principal dans la prochaine version de MacOS X. Lorsque nous nous sommes intéressés à MacOS X, il n'existait pas de client NIS fournit par Apple. Celui-ci a été intégré dans une version récente, mais il ne permet de récupérer que les informations utilisateurs (la table `passwd`) et pas les autres comme les montages ou les imprimantes. Nous n'avons donc pas avancé plus avant sur ce point, en considérant par ailleurs que NIS serait remplacé à terme par LDAP.
L'autre client disponible était donc LDAP et comme nous avions déjà un annuaire OpenLDAP nous avons regardé comment l'étendre pour qu'il fonctionne avec MacOS X.
LDAP peut être utilisé pour 2 fonctions :
• l'authentification et la gestion des utilisateurs (nom de login, mot de passe, répertoire utilisateur, uid, groupes…),
• la gestion des autres ressources disponibles (répertoires partagés, imprimantes…).
Pour stocker l'ensemble de ces informations dans un annuaire, il faut un schéma : Apple propose plusieurs schémas et mappages d'attributs et d'objets LDAP vers des objets propres à MacOS X.
OpenDirectory est un schéma propre à Apple fonctionnant sur MacOS X serveur. En particulier, il définit les utilisateurs avec une richesse en termes d'attributs bien supérieure à celle des comptes utilisateurs UNIX traditionnels.
Nous avons préféré utiliser pour la partie comptes utilisateurs le schéma basé sur le RFC 2307 [2] qui suffit, en termes d'objets et d'attributs, pour configurer tout ce qui est nécessaire dans un environnement UNIX. Ces informations sont définies dans `nis.schema`. Pour le moment les seules autres ressources utilisées sont les montages de répertoires distants, qui sont définis dans `apple.schema`. Ces schémas sont dans `/etc/openldap/schemas`.

5.1Les objets et attributs

Voici les objets de ces schémas que nous utilisons, le nom des attributs est suffisamment éloquent :
objectclass (1.3.6.1.1.1.2.0 NAME '**posixAccount**' SUP top AUXILIARY
DESC 'Abstraction of an account with POSIX attributes'
MUST (cn $ uid $ uidNumber $ gidNumber $ homeDirectory)
MAY (userPassword $ loginShell $ gecos $ description))

objectclass (1.3.6.1.1.1.2.2 NAME '**posixGroup**' SUP top STRUCTURAL
DESC 'Abstraction of a group of accounts'
MUST (cn $ gidNumber)
MAY (userPassword $ memberUid $ description))

objectclass (
1.3.6.1.4.1.63.1000.1.1.2.8
NAME '**mount**'
SUP top STRUCTURAL
MUST (cn)
MAY (mountDirectory $
mountType $
mountOption $
mountDumpFrequency $
mountPassNo))

Les fichiers de configuration MacOS X relatifs aux annuaires sont stockés dans `/Library/Preferences/DirectoryService/` :

- `DSLDAPv3PlugInConfig.plist` : qui contient le "mappage" des attributs LDAP vers ceux MacOS X au format XML,
- `SearchNodeConfig.plist` : qui contient les types d'authentification supportées et l'adresse du serveur LDAP au format XML.

5.2La sécurité de LDAP

L'authentification via un annuaire LDAP n'est pas très sécurisée puisque le mot de passe est envoyé en clair du client LDAP vers le serveur. C'est celui-ci qui authentifie ou non l'utilisateur. Pour améliorer la sécurité il est possible d'utiliser LDAPS (LDAP sur SSL) avec un certificat serveur. Dans ce cas, on éviterait au moins le passage du mot de passe en clair sur le réseau. Ce "trou" de sécurité existe de toute manière au laboratoire en particulier pour ce qui concerne les connexions vers les serveurs POP ou IMAP.
Une meilleure solution consisterait à utiliser un serveur LDAP avec une authentification Kerberos. Celui-ci assure une authentification unique auprès d'un serveur d'authentification et fournit des jetons permettant d'accéder à des serveurs sécurisés. D'après Apple, Panther, la prochaine version de MacOS X (10.3), intègrerait mieux Kerberos en particulier dans le client LDAP. Les premiers tests effectués avec une version provisoire de Panther ont permis de voir que le client LDAP (basé sur openldap) était bien compilé avec le support Kerberos.

6 Partage de fichiers

L'ensemble des fichiers du laboratoire (répertoires utilisateurs et programmes) est accessible par automontage. Les tables d'automontage sont stockées dans des tables NIS+ utilisables par des clients NIS+ (Solaris) et NIS (Linux, BSD, HP/UX). L'automontage permet une souplesse importante dans la gestion des systèmes de fichiers et une optimisation des montages nécessaires. L'accès aux répertoires partagés est aussi facilité, puisqu'il se fait de manière transparente pour l'utilisateur. Les machines Windows font aussi de l'automontage à leur manière puisque les répertoires utilisateurs sont montés au login, et que les répertoires partagés sont montés à l'accès. La seule différence est qu'il n'existe pas de table unique d'automontage bien qu'elle puisse être mise en place dans Active Directory.
Nous présenterons ici uniquement le montage des répertoires utilisateurs, mais le principe peut être étendu aux autres répertoires ou systèmes de fichiers partagés.
Il existe donc une table NIS+ indirecte `auto_home` qui contient la description de l'ensemble des répertoires utilisateurs des membres du laboratoire. Cette table est montée sur `/home`. Elle contient en fait 2 éléments :

- le point de montage (le nom d'utilisateur `user`),
- le répertoire à monter : `serveur:/répertoire`.

L'accès à `/home/user` effectue le montage.
L'intérêt d'une telle table est évident. Le répertoire utilisateur est indépendant du client et du serveur : c'est toujours `/home/user`. Nous souhaitions avoir exactement la même chose sous MacOS X.

6.1L'automontage : automount et login

Tout d'abord il existe 2 automonteurs sous MacOS X dont un seul est réellement supporté : **automount**. Amd existe mais n'est pas utilisé. Automount est en fait le monteur de MacOS X pour tout ce qui est réseau (NFS , CIFS, AFP). Il a deux gros défauts :

- aucune documentation si ce n'est une aide succincte (tapez `automount -m`),
- il est totalement propriétaire et développé par Apple (le code source est cependant disponible).

Sinon il peut lire les tables de montage à partir de sources variées dont le service DirectoryServices, c'est-à-dire l'ensemble des annuaires auxquels le Mac a accès. On pourra donc stocker les tables de montage dans l'annuaire LDAP comme pour les autres informations utilisateurs.
Tous les montages réseaux autres que ceux effectués manuellement par l'utilisateur sont réalisés par automount. Cela signifie qu'il n'existe pas de montage statique, et que tous sont dynamiques. D'autre part automount ne supporte que les maps directes mais son fonctionnement est un peu particulier (voir 6.1.1).
Il existe aussi un autre automonteur complètement caché celui-ci intégré au login qui permet de monter le répertoire utilisateur, un peu à la mode Windows. L'avantage de cet automonteur est que le système lui passe le nom d'utilisateur et le mot de passe si c'est nécessaire. On peut donc monter des volumes CIFS ou AFP automatiquement, sans interaction avec l'utilisateur.
Il y a donc 2 façons avec MacOS X de réaliser les montages de répertoires utilisateurs via un annuaire, soit :
- décrire le montage explicitement en utilisant un objet de type mount (en fait cela revient à créer une table `auto_home` dans LDAP),
- décrire le montage en utilisant un attribut explicite au format XML dans un objet de type PosixAccount. Cet attribut est spécifique à MacOS X. Le format XML de cet attribut ne permet pas de passer les options au montage.

6.1.1Configuration et fonctionnement d'automount

Automount peut utiliser des objets LDAP du type mount dont voici le format :

```
dn: cn=gailuron:/users1/blain,dc=laas,dc=fr
objectclass: mount
cn: gailuron:/users1/blain
mountDirectory: /home/blain
mountType: nfs
mountOption: -P
mountOption: -T
```

Les noms des attributs sont assez clairs :
Cn : répertoire à monter,
MountDirectory : point de montage
MountType : protocole, peut être NFS, SMB ou AFP
MountOption (multivaluées) : options

Si on utilise ces objets pour le montage des répertoires utilisateurs, il faut créer une entrée LDAP par utilisateur. On y ajoutera aussi tous les répertoires partagés que l'on souhaite pour des applications ou des données.
Au lancement, automount crée un lien symbolique entre le point de montage et la même arborescence sous `/automount` s'il n'existe pas. Cela signifie que si on crée une nouvelle entrée dans l'annuaire et qu'on veut y accéder, il faut vider le cache local de l'annuaire (`lookupd -flushcache`) et relancer automount. Cela revient à redémarrer la machine.

6.1.2Configuration de l'automonteur de login

L'automonteur de login utilise un attribut ldap dans les objets décrivant le compte utilisateur. C'est un attribut ajouté par une classe différente à la classe posixAccount qui provient du schéma de chez Apple :

```
dn: uid=blain,dc=laas,dc=fr
objectClass: aplMacOSXUser
apple-user-homeurl:
<home_dir><url>smb://gailuron.laas.fr/home</url><path>blain</path></home_dir>
```

apple-user-homeurl contient au format XML un URL du répertoire partagé. Il sera monté sur le répertoire utilisateur. Le reproche que l'on peut faire à cet automonteur est que la documentation sur le format y est inexistante et que, de plus, il est impossible de passer des options par un URL.

6.2NFS vs CIFS

Dans un premier temps, nous pensions que l'utilisation de CIFS plutôt que de NFS résoudrait un problème majeur de sécurité. En effet, sur les ordinateurs personnels du type PC ou Mac les utilisateurs ont l'habitude d'être administrateurs de leur machine, c'est-à-dire qu'ils peuvent installer les applications ou matériels qu'ils souhaitent. Il est difficile de changer les habitudes. D'autre part, étant donné le nombre important d'applications existantes pour ces systèmes, l'équipe système ne peut pas passer son temps à installer de nouveaux logiciels. Enfin, il est quasiment impossible de gérer les portables sans déléguer le droit d'administration afin que ceux-ci puissent être reconfigurés si nécessaire sur un autre site. Ces différentes raisons ont fait que, sur certains postes, il a fallu donner des privilèges au propriétaire de la machine.
Sur un PC cela ne pose pas trop de problèmes en terme de sécurité de par l'utilisation du protocole CIFS. En effet, pour effectuer un montage CIFS un utilisateur a besoin d'être authentifié formellement par le serveur de fichiers ou un serveur d'authentification: il doit fournir, à un moment ou à un autre, son nom d'utilisateur et son mot de passe. Sans eux, il ne peut effectuer de montage. Donc les droits sur des volumes distants sont toujours les droits de cet utilisateur et non pas ceux d'un autre. Pour se faire passer pour un autre, il faudrait connaître son mot de passe. La sécurité de NFS est plutôt basée sur la notion de confiance de machine à machine, même si l'authentification utilise aussi l'uid de l'utilisateur sur le

poste client. NFS considère que lorsqu'une requête arrive d'une machine autorisée, il peut avoir confiance dans l'identité de l'utilisateur contenu dans la requête. Au LAAS, la liste des machines de confiance est dans un netgroup : toute machine appartenant à ce netgroup peut faire des montages des répertoires applicatifs et utilisateurs. Un utilisateur qui est root sur une machine peut sans problème se faire passer localement pour un autre utilisateur (par exemple en utilisant la commande su), mais plus grave il peut effectuer un montage et donc se faire passer pour cet autre utilisateur sur un serveur NFS distant. En clair, il est impossible de protéger correctement les fichiers utilisateurs dans un environnement NFS où les utilisateurs peuvent être root sur leurs machines. Il n'existe alors que 4 solutions pour introduire des machines UNIX dans ce type de réseau :

- ne pas donner le mot de passe root aux utilisateurs (c'est le cas pour les stations Sun),
- mettre la machine dans le netgroup des machines de confiance si le propriétaire est de confiance (cas de certains PC sous Linux),
- utiliser une version de NFS où l'utilisateur doit être authentifié par un serveur de confiance pour que ses requêtes NFS puissent être acceptées, comme par exemple dans NFS V4 Kerberizé (ne fonctionne que sous Solaris),
- utiliser un autre protocole d'accès à des répertoires distants (CIFS, AFS).

Pour les postes MacOS X dont les utilisateurs veulent conserver les droits d'administration, la quatrième solution nous paraissait la meilleure. Il existait déjà des serveurs de fichiers CIFS (Samba) permettant l'accès à son répertoire utilisateur pour tous les PC sous Windows, et MacOS X intègre en standard mount_smbfs : un logiciel permettant le montage et l'accès à des partages SMB/CIFS. De plus comme il a déjà été indiqué, l'automonteur du login peut passer nom d'utilisateur et mot de passe si nécessaire, comme dans le cas de CIFS.

6.2.1Configuration de CIFS

Sous LDAP, pour utiliser CIFS, nous avons utilisé l'automonteur de login qui permet de passer le mot de passe. Et à chaque compte utilisateur nous avons donc rajouté un attribut :

```
apple-user-homeurl:
<home_dir><url>smb://gailuron.laas.fr/home</url><path>blain</path></home_dir>
```

qui décrit le répertoire distant à monter.
Un autre attribut décrit le point de montage local : le répertoire utilisateur.

```
HomeDirectory: /home/blain
```

6.2.2Test de CIFS

Hélas le client CIFS (mount_smbfs) ne fonctionne pas correctement. Il y a de gros problèmes à l'ouverture de gros fichiers, lors du transfert, et les performances sont médiocres. De plus, il ne permet pas de gérer correctement les droits utilisateurs comme à partir d'un PC Windows NT.
Et point critique, lorsqu'un autre utilisateur se connecte sur la machine, soit à partir d'une session distante via ssh, soit à la console après l'utilisateur précédent, et si celui-ci a fait un montage CIFS, le nouvel utilisateur a accès à ce montage avec exactement les mêmes droits que le premier. Il n'y a pas de contrôle en fonction de l'uid. Ceci empêche toute utilisation fiable de CIFS.
Comme il n'existe pas de serveurs AFP sur nos machines UNIX (cap/aufs et netatalk sont arrêtés depuis longtemps), le seul moyen d'accès reste NFS malgré le problème important indiqué ci-dessus. Pour le moment les utilisateurs n'ont pas le droit d'administration de ces machines.

6.3NFS

Voici comment est configuré au LAAS, le montage des répertoires utilisateurs par NFS. Nous n'avons pas utilisé l'automonteur de login. En effet, si cette méthode est la plus simple à mettre en place, il y est impossible de passer l'option –P qui permet une connexion en utilisant un port privilégié réservé. Or, au LAAS, seul ce type de connexion NFS est autorisé. De plus, si on veut mettre en place d'autres options telles que des connexions TCP, c'est impossible. Donc cette solution est inutilisable dans notre cas.
Voici cependant comment il faudrait procéder. Il suffit de rajouter un attribut de cette forme-là à chaque entrée utilisateur sous LDAP :

```
apple-user-homeurl:
<home_dir><url>nfs://gailuron.laas.fr/users1</url><path>blain</path></home_dir>
```

Une remarque à ce propos : il est tout à fait possible d'effectuer un montage manuel pour n'importe quel utilisateur en utilisant dans le Finder la commande « Se connecter à un serveur ». On peut effectuer un montage AFP, CIFS ou NFS. Il faut noter que le montage se passe alors comme le montage du login en donnant un URL et qu'il est donc impossible de passer des options.

6.3.1Le problème de la corbeille

Le premier choix que nous avions fait était de monter les répertoires utilisateurs dans `/home/user`, où user est le nom d'utilisateur. Ceci fonctionne bien sur Solaris, mais aussi sur tous les autres UNIX installés au laboratoire. Cela permet à un répertoire utilisateur d'être identifié de manière indépendante du serveur sur lequel il réside.
Pour cela on crée donc l'équivalent de auto_home sous LDAP avec un montage par répertoire utilisateur de la forme :

```
dn: cn=gailuron:/users1/blain,dc=laas,dc=fr
objectclass: mount
cn: gailuron:/users1/blain
mountDirectory: /home/blain
mountType: nfs
mountOption: -P
```

Mais si le montage s'effectue sans problème sous MacOS X, la corbeille du Finder ne fonctionne plus. Si on essaie d'effacer un fichier de la racine du répertoire utilisateur une erreur est affichée. Et si on essaie d'effacer un fichier d'un sous-répertoire il est directement effacé sans passer par la corbeille. Il semble que ce problème soit dû au fait que le répertoire au-dessus n'est pas sur le même système de fichiers : `/home/user` est sur le serveur distant et `/home` sur le disque système local.
La solution a donc consisté à faire le montage des répertoires contenant les répertoires utilisateurs dans `/Network/Users/serveur/répertoire` pour tous les serveurs. On crée donc sous LDAP un ensemble d'objets mount de la forme :

```
dn: cn=gailuron:/users1,dc=laas,dc=fr
objectClass: mount
cn: gailuron:/users1
mountType: nfs
mountDirectory: /Network/Users/gailuron/users1
mountOption: -P
```

Ensuite il faut modifier les objets utilisateurs. L'attribut homeDirectory contient `/home/user` qui est bien le répertoire utilisateur sur tous les autres UNIX. Nous avons donc ajouté un autre attribut, aplHomeDirectory, en utilisant un schéma fournit par Novell. Cet attribut contient le répertoire utilisateur sous MacOS X : `/Network/Users/serveur/répertoire/user`. Il faut ensuite modifier, par l'intermédiaire du Gestionnaire de Répertoires, la configuration LDAP pour mapper cet attribut vers l'attribut local à MacOSX définissant le répertoire utilisateur : NFSHomeDirectory.
Voici donc le nouveau contenu d'un compte utilisateur sous LDAP :

```
dn: uid=blain,dc=laas,dc=fr
objectClass: posixAccount
objectClass: aplMacOSXUser
cn: Laurent Blain
uid: blain
uidNumber: 3913
gidNumber: 39
gecos: Laurent Blain,A136b,7809
loginShell: /usr/local/bin/tcsh
homeDirectory: /home/blain
aplHomeDirectory: /Network/Users/gailuron/users1/blain
```

Une fois la configuration entièrement spécifiée, elle peut être dupliquée sur d'autres postes en copiant les fichiers `DSLDAPv3PlugInConfig.plist` et `SearchNodeConfig.plist`.

6.3.2Problème de la mise en veille

Un autre point très ennuyeux est la mise en veille, activée par défaut sur MacOS X. Celle-ci entraîne des dysfonctionnements de NFS lors de la sortie de la mise en veille. Il semble y avoir des problèmes avec le code NFS intégré au noyau qui change de mode de fonctionnement lors de la sortie de la mise en veille. Celle-ci a été désactivée sur les Mac.

6.3.3Unicode

Un autre point qui nous gêne est que le codage des caractères pour les noms de fichiers utilisé par MacOS X n'est pas celui de nos serveurs et stations Solaris. Dans le premier cas c'est Unicode, UTF-16 pour HFS+ et UTF-8 pour NFS, dans le second c'est ISO-Latin-1. Les noms contenant des caractères accentués apparaissent donc différemment sous Sun et sous Mac. La solution consiste, mais ce n'est pas facile, à faire évoluer les systèmes de fichiers sur Sun vers UTF-8. Ceci se fera sans doute progressivement.

7 Gestion des applications

Dans notre configuration réseau nous essayons le plus possible d'installer les applications sur des répertoires partagés pour éviter d'avoir à les installer sur chaque poste et pour pouvoir faire évoluer les versions plus facilement. Sur les stations UNIX, le répertoire `/usr/local` est dédié à cela. Une table d'automontage indirecte auto_usr_local montée sur `/usr/local` permet d'accéder aux applications.
Dans le monde Windows, le nombre d'applications pouvant être exécutées en mode partagé est très réduit en particulier à cause du registre, mais aussi d'une mauvaise gestion des droits. Mais un certain nombre est installé sur un serveur et accessible en automontage.
Nous avons donc étudié l'installation d'applications partagées dans le monde MacOS X.

7.1Applications locales

Certaines applications sont installées localement de manière à optimiser les performances et pour éviter certains dysfonctionnements. D'autres nécessitent une modification du système. Et bien sûr, pour les portables, les applications sont installés localement.
Quelques applications modifiant le système sont :
• OSXVNC(www.redstonesoftware.com),
• le serveur X11 de Apple.

7.2Applications partagées MacOS X

Les applications partagées sont installées dans un répertoire sur un serveur NFS. Ce répertoire est automonté dans `/Network/Applications/softs_MacOSX`. Nous avons défini une entrée mount dans LDAP comme pour les répertoires utilisateurs.
Une application MacOS X est composée d'un dossier avec une extension .app. Son installation, dans le cas où le système n'a pas besoin d'être modifié est simple : il suffit de copier ce dossier dans le répertoire où l'on veut mettre l'application. Il faut utiliser la commande `CpMac` (dans `/Developer/Tools`) pour gérer correctement Data et Ressources lors de la recopie.

On peut donc installer une application sur n'importe quel répertoire local ou distant. C'est en tout cas vrai théoriquement. Pour le moment nous avons réussi à faire fonctionner quelques applications :
• Mozilla 1.4.1,
• Eudora 6,
• le client ICA de Citrix.

Quelques applications posant problème sont :
• Microsoft Office X,
• Netscape 7.

7.3Applications UNIX : automonteur /usr/local

Pour les applications "UNIX", il est très facile de faire comme pour les Sun en créant sous `/usr/local` des points d'automontage. Il existe par défaut des répertoires `bin`, `lib`, `share`, `include` et `man` qui peuvent rester locaux à la machine.

7.4Mise à jour

La mise à jour des applications peut se faire simplement si celles-ci ne nécessitent pas de modifications de l'OS :
• si ce sont des applications partagées, il suffit d'installer la nouvelle version sur le serveur,
• si c'est une application locale, il suffit de copier à distance la nouvelle version en utilisant `scp`.
La mise à jour de l'OS aussi peut se faire facilement à distance en utilisant la commande `softwareupdate`. Il est même possible de choisir quels packages installer en l'intégrant dans un script.

8 Les Imprimantes

Il n’y a pas de solution simple sous Solaris pour installer des imprimantes et les rendre visibles et accessibles à l’ensemble des utilisateurs et des applications. Par exemple Mozilla nécessite l’installation d’un serveur Xprint, et StarOffice implique que chaque utilisateur configure ses imprimantes en utilisant l’outil spadmin. Et les autres applications utilisent le système d’impression BSD.
Les Mac se basent aussi sur un ensemble de protocoles de découverte et d’accès aux imprimantes, mais ensuite les applications MacOS X reconnaissent automatiquement et peuvent utiliser ces imprimantes et l’ensemble de leurs options. Parmi ces protocoles, notre intérêt s’est surtout porté sur CUPS [3].

8.1CUPS

Pour pouvoir utiliser CUPS, nous avons choisi d’installer un serveur CUPS sur une machine Solaris. Ce serveur rend immédiatement disponible et visible à l’ensemble des clients CUPS, MacOS X ou Linux, les imprimantes qui y sont déclarées. CUPS utilise des PPD pour décrire les imprimantes et permet donc facilement d’offrir toutes les options de ces dernières. De cette manière, l’utilisateur n’a rien à configurer sur sa machine mais uniquement à choisir son imprimante.
Il y a quelques reproches à faire à CUPS, en particulier, le fait qu’il n’y a pas de distinction claire entre le serveur et le client. C’est en configurant le fichier `cupsd.conf` qu’on limite les fonctions à l’aspect client ou qu’on le transforme en serveur. Dans MacOS X, c’est en cochant la case Partager des Imprimantes qu’on active le serveur.

Voilà comment cela se traduit dans `cupsd.conf` :

Mode client seul	Mode serveur
ServerName 127.0.0.1 #Port 631 Listen 127.0.0.1:631 #BrowseAddress @LOCAL BrowseInterval 0	#ServerName hostname Port 631 #Listen 127.0.0.1:631 BrowseAddress @LOCAL BrowseInterval 30 Allow From @LOCAL

Le problème est qu'il semble que certains Linux installent par défaut CUPS en mode serveur et surtout qu'ils diffusent toutes les imprimantes qu'ils connaissent, même les imprimantes réseaux. On retrouve la même propriété sur MacOS X si on active le partage d'imprimantes. Donc toute machine, Linux ou MacOS X, qui possède une imprimante locale et qui souhaite la partager, va aussi partager toutes les imprimantes réseaux qu'elle connaît, par exemple celles auxquelles elles accèdent par Appletalk ou lpd. Et sur chaque Mac, on va lister un nombre d'imprimantes considérable dont certaines plusieurs fois, puisqu'on verra toutes celles disponibles sur tous les serveurs. Une solution serait de limiter le client CUPS à la consultation d'un seul serveur. C'est théoriquement possible mais n'a pas fonctionné dans notre configuration.

8.2Rendez-vous et AppleTalk

Il existe aussi d'autres protocoles d'impression dont Rendez-vous qui est relativement nouveau et AppleTalk qui est historiquement le protocole réseau d'Apple. Rendez-vous ne nous apporte rien, car il est basé sur l'auto-découverte des matériels, et il n'a d'intérêt que dans un réseau sans DNS et annuaire centralisé. AppleTalk est installé par défaut sur toutes les imprimantes réseau et permet à ces dernières de s'auto-déclarer, et donc à n'importe quel Mac de les voir et d'y imprimer. AppleTalk est cependant un protocole en fin de vie (il est d'ailleurs désactivé par défaut sur certaines versions de MacOS X).

Une fois l'authentification, NFS et le système d'impression CUPS choisis et configurés sous MacOS X, voici la configuration réseau clients/serveurs que l'on devrait avoir au LAAS :

	Client UNIX	***Client Windows***	***Client MacOS X***
Authentification	Serveur LDAP Solaris	Serveur Active Directory Windows	Serveur LDAP Solaris
Fichiers	Serveur NFS Solaris	Serveur CIFS Solaris	Serveur NFS Solaris
Applications	Serveur NFS Solaris	Serveur CIFS Windows	Serveur NFS Mac OX Serveur
Imprimantes	Serveur LPD Solaris	Serveur CIFS Windows	Serveur CUPS Solaris

9 L'administration distante

L'administration à distance des Mac est une fonction essentielle pour pouvoir les gérer correctement. Une des grandes difficultés du monde Windows est justement le fait que les PC ne sont pas accessibles facilement à partir d'autres OS. On distinguera 2 moyens d'intervention :

- celui classique d'ouverture d'une session en mode terminal sur la machine distante (login distant classique sous UNIX),
- celui d'une prise de contrôle à distance de l'écran et de la souris, essentiels pour dépanner dans un environnement multi-fenêtré.

9.1SSH

Par défaut, le protocole de connexion à distance sur MacOS X est ssh, ce qui nous convient très bien en terme de sécurité. Pour activer le service nous passons simplement la variable SSHSERVER à YES dans `/etc/hostconfig`.
Pour permettre le lancement de scripts distants sans avoir à taper de mot de passe, ce qui serait rébarbatif, nous autorisons l'authentification par clé publique (PubkeyAuthentication dans sshd_config) en donnant la clé du serveur principal du laboratoire.
À partir de là il est très facile d'écrire des scripts permettant d'intervenir à distance sur un ou plusieurs Mac. Certaines opérations sont accessibles par ligne de commande. Par exemple, l'ensemble de la configuration initiale des Mac après une installation standard est réalisée par un ensemble de scripts : configuration LDAP, SSH, installation de logiciels. D'autres nécessitent encore de travailler sur la console.

9.2VNC

VNC, dans la version RealVNC (www.realvnc.com), est l'outil que nous utilisons tous les jours pour gérer à distance nos PC sous Windows à partir d'autres PC ou de stations UNIX. Il existe aussi une version de VNC sur MacOS X, que nous installons systématiquement. Elle nous permet d'intervenir aisément pour dépanner les utilisateurs. La version installée est celle de Redstone software (www.redstonesoftware.com), OSXVNC, qui n'intègre qu'un serveur. Nous ne déployons pas le client systématiquement sur tous les postes.

9.3Scripter et automatiser les tâches

Un des problèmes des environnements multi-fenêtres est leur accès souvent unique en mode "cliquodrome" ce qui rend les tâches d'administration difficiles à scripter et automatiser. C'est le cas en particulier sous Windows où nous avons cependant pu au moins automatiser l'installation et la configuration via quelques scripts simples. MacOS X offre des langages de scripts qui nous sont plus familiers, Shell et Perl, mais a aussi une interface fortement cliquodrome. Nous avons donc commencé à regarder si toutes les tâches pouvaient être scriptées. Cela nécessite deux choses :

• d'une part des fichiers configurations facilement manipulables par script (du texte),
• d'autre part des commandes en mode ligne pour toute exécution de programme MacOS X.

9.3.1Les fichiers de configuration

Les fichiers de configuration sous MacOS X sont de 2 types. Soit ce sont des fichiers classiques BSD, tels que `/etc/passwd`, que l'on connaît et que l'on sait manipuler. Soit, et ce sont les plus nombreux, ce sont des fichiers au format XML, non documentés, mais heureusement au format texte. Les deux fichiers de configuration `DSLDAPv3PlugInConfig.plist` et `SearchNodeConfig.plist` qui décrivent la configuration des annuaires et de LDAP sont dans ce format. En regardant le contenu de ces fichiers on peut comprendre comment modifier certaines valeurs, mais la structure et la liste des attributs ne sont décrites nulle part. Il faut donc travailler de manière empirique. Ensuite une simple manipulation par sed permet de le modifier sans problème.

9.3.2Les commandes

La mise à jour automatique est le bon exemple d'une application utilisable par les deux types d'interface. Elle apparaît d'abord dans les Préférences Systèmes sous le nom "Mise à jour de logiciels" et permet une mise à jour automatique de l'OS et des applications principales d'Apple. L'interface est donc du type "cliquodrome".
Mais cette fonction peut aussi être appelée par la commande en ligne `softwareupdate`. Celle-ci peut être intégrée dans un script de manière à automatiser la mise à jour tout en choisissant ce que l'on veut installer. Ce double choix est très appréciable.
Un ensemble d'outils sont mis à la disposition des administrateurs pour travailler avec des scripts. Nous en citerons quelques-uns :

• `CpMac` : permet de copier des fichiers avec toutes les informations MacOS (plutôt que cp),
• `MvMac` : permet de déplacer des fichiers avec toutes les informations MacOS (plutôt que mv),
• `open Application.app` : lance une application multi-fenêtrée en tâche de fond,
• pour lancer une application MacOS X à partir de la ligne de commande, il suffit de taper :
`/Path to Application.App/Contents/MacOS/Application.`

Nous utilisons ces facilités dans le script de configuration de MacOS X. Ce script configure l'authentification LDAP, le mappage LDAP/MacOS X, le serveur SSH, et installe des applications en copiant simplement répertoires et fichiers.

9.3.3L'installation réseau

Nous n'avions pas, au moment de rédiger cet article, de serveur MacOS X permettant de réaliser simplement des installations réseaux. A priori ces fonctionnalités existent et il "ne reste" qu'à les tester pour évaluer leur difficulté de mise en œuvre. Le serveur devant être mis en place d'ici peu, nous espérons pouvoir donner quelques éléments lors de la présentation aux JRES. Une fois ce service mis en place, on pourra considérer que MacOS X s'intègre parfaitement et au même niveau que les postes de travail sous Solaris et Windows dans le réseau du laboratoire.

10Les limites

On ne peut pas dire que le système MacOS X soit parfait et qu'il fournisse tous les services demandés. En plus des quelques points évoqués ci-dessus, il faut indiquer d'autres limites que nous avons notées.

10.1Les portables

MacOS X est un système très utilisé sur des portables, PowerBook et Ibook, comme Windows. Nous avons indiqué que notre architecture a été pensée et élaborée à une époque où ceux-ci étaient quasi inexistants, et donc que tout poste de travail est censé être connecté au réseau pour pouvoir accéder aux ressources partagées. Le problème s'est d'abord posé avec les nombreux portables sous Windows. Cependant 2000 et XP permettent de s'affranchir de la connexion au réseau grâce à un système de cache :

• un cache pour l'authentification qui permet d'éviter de créer des comptes locaux sur les machines et qui fait que l'utilisateur conserve le même environnement avec ou sans réseau,
• un cache pour les fichiers, les fichiers hors-connexion, qui garde une copie locale des fichiers ou répertoires les plus accédés ou mis en cache manuellement : les fichiers sont synchronisés à la reconnexion au réseau.

Pour le moment aucune de ces deux fonctionnalités n'existe sous MacOS X. Lorsqu'un portable en configuration LAAS, connecté à annuaire LDAP et réalisant des montages NFS, démarre sans réseau, cette phase est très longue : le portable

essayant pendant un bon moment d'accéder aux ressources. Il semble que la détection de l'absence de réseau ne soit pas parfaite. On retrouve le même problème de gestion du réseau lors de la mise en veille et en sortie de celle-ci, les connexions NFS défaillant.
S'il n'y a pas pour le moment de solution pour gérer un cache des utilisateurs, nous sommes en train d'étudier des logiciels pour gérer la synchronisation de fichiers et répertoires.

10.2Applications MacOS9

Dans certains cas, nous sommes hélas obligés d'utiliser des applications MacOS 9 à faire exécuter en environnement Classic (boîte de compatibilité MacOS 9) : soit parce que l'application n'existe pas sous MacOS X, soit parce que la version MacOS X n'a pas les fonctionnalités désirées. C'est par exemple le cas de Netscape 4 qui est la seule version de Netscape qui fonctionne avec LabIntel. Le problème est que la gestion de ces applications nécessite une administration de MacOS 9 et que nous perdons les avantages de MacOS X. On peut espérer cependant que les applications évolueront et que Classic ne sera plus nécessaire.

11Conclusion

MacOS X semble un système prometteur à plusieurs points de vue. C'est un système UNIX avec tout le bien qu'on peut en penser et la connaissance et la maîtrise qu'on en a. Il offre à la fois tous les outils de développement GNU connus, et la possibilité d'y installer et utiliser les applications UNIX/X11 classiques. Il a aussi toute la richesse applicative du monde MacOS et une interface utilisateur très élaborée. Notre étude pour savoir s'il était possible de l'administrer et de l'intégrer facilement dans un réseau comme celui du LAAS nous a conduit à répondre positivement à cette question. Il y a encore un manque d'information sur certains aspects du système qui nous pousse à travailler de manière empirique, et quelques manques ou dysfonctionnements que l'on espère se voir régler dans les prochaines versions du système, mais l'approche semble prometteuse. Le choix du support de protocoles reconnus comme NFS, CIFS, LDAP, NIS sans trop de modifications spécifiques à Apple permet une intégration relativement aisée. Et la double interface, ligne de commandes ou fenêtrée, pour accéder au système UNIX est vraiment très complète et permet une administration simple et automatisable.

Références

[1] Apple, "Inside Mac OS X : System Overview", Février 2003
[2] RFC 2307,"An Approach for Using LDAP as a Network Information Service", Mars 1998
[3] CUPS,"Software Administrators Manual, CUPS v1.1.19", 2003