Annuaire LDAP ou SGBD : quelles solutions pour un référentiel unique

Jacques Eudes, Gérard Forestier, Eric Payan

Annuaire LDAP ou SGBD : quelles solutions pour un référentiel unique ?

J. Eudes, ***Jacques.Eudes@ujf-grenoble.fr***

G. Forestier, ***Gerard.Forestier@imag.fr***

E. Payan, ***Eric.Payan@univ-lyon1.fr***

1. Introduction

La multiplication des applications recourant à une authentification et l'émergence de projets transversaux aux composantes des établissements a rendu difficile la gestion des droits d'accès informatiques avec les outils traditionnels.
Face à ce constat la nécessité de déployer des outils permettant une identification unifiée des étudiants, des enseignants et du personnel administratif et technique sur toutes les ressources informatiques est devenue une nécessité absolue.
Dans cet article nous exposons l'expérience que nous avons menée à l'Université Joseph Fourier Grenoble 1 (UJF). Dans la section 2 nous présentons le déploiement, au sein de l'UJF, de l'annuaire LDAP et le développement des outils associés. Le projet régional AGALAN fait suite à ces travaux, il est présenté dans la section 3. Nous discuterons ensuite des annuaires et des SGBD dans la section 4 et 5 puis nous tenterons de proposer des solutions d'intégration des annuaires dans la section suivante. En annexe se trouvent les résultats des tests effectués sur différents annuaires LDAP.

2. LDAP au sein de l'Université Joseph Fourier (Grenoble 1) depuis 1999

2.1 Le projet

Le déploiement d'un annuaire LDAP au sein de l'Université Joseph Fourier de Grenoble a été entrepris depuis le printemps 1999 sur la base d'un cahier des charges simple :« Savoir reconnaître les nôtres ».
L'authentification unifiée des étudiants, des enseignants et des personnels administratifs et techniques sur toutes les ressources informatiques :

- serveurs pédagogiques,
- messagerie,
- formations en ligne,

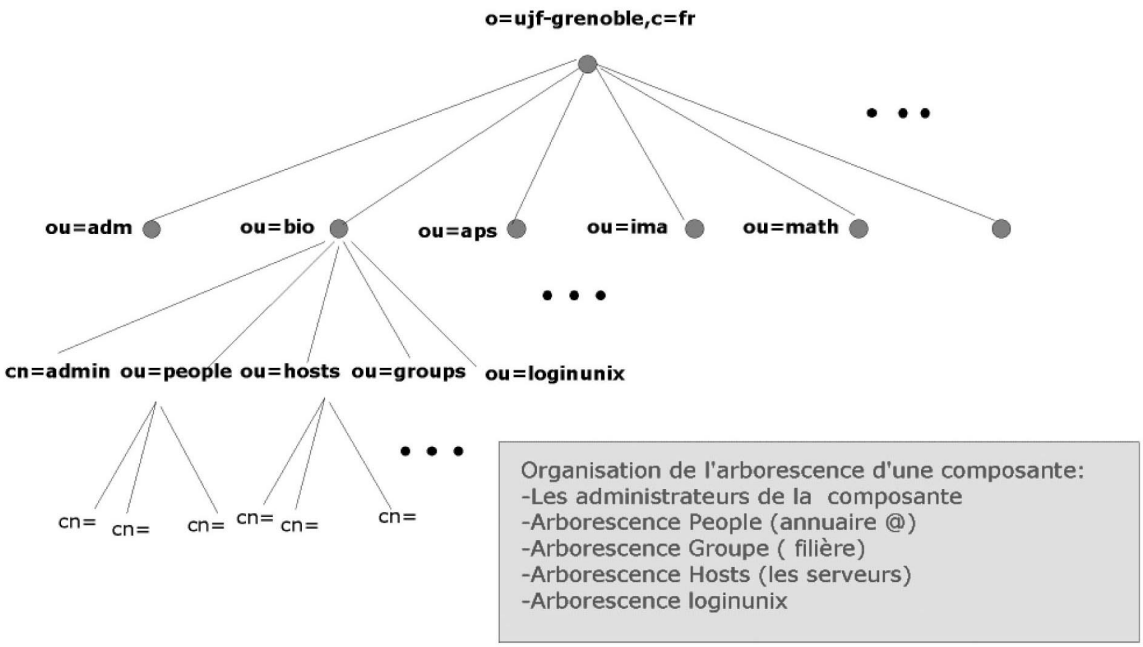
constituait la motivation majeure de ce projet.

La nécessité d'une gestion coopérative du contenu de l'annuaire par les personnels de chacune des composantes de notre établissement nous a naturellement conduit à nous tourner vers une solution basée sur un annuaire LDAP.

2.1.1 Sa réalisation

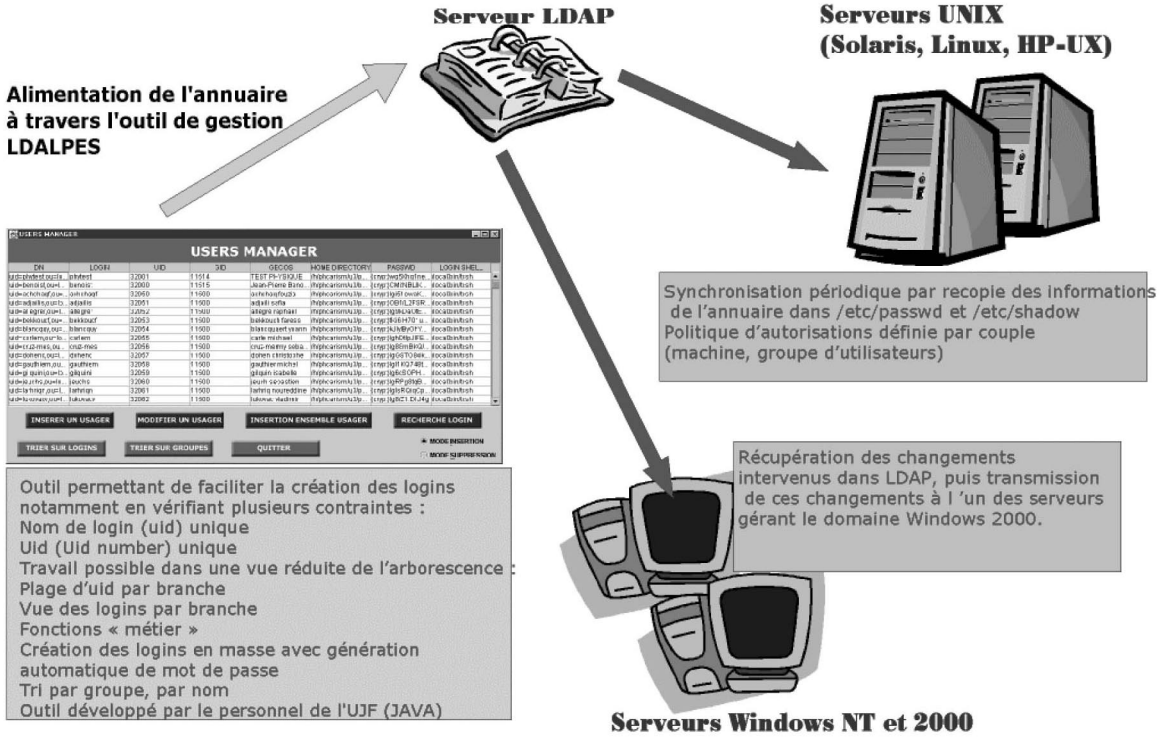
Ce projet a été structuré en 3 parties développées durant l'été 99. Ces parties sont :

La mise en route d'un serveur d'annuaire (logiciel SUN abandonné par l'éditeur depuis), et la définition de la structure de son arborescence.
La structure choisie pour cet arbre s'appuie directement sur le découpage de notre établissement en composantes pour l'enseignement et en laboratoires pour la recherche en créant une branche pour chacun (voir figure ci-dessous). Cette structure simplifie les règles d'accès car elle permet d'associer un administrateur à chacune des branches et donc de déléguer l'administration.



Le développement de logiciels clients permettant la synchronisation des logins sur les machines clientes. Le choix d'une stratégie de recopie des logins sur les machines clientes à intervalles réguliers a permis de supporter une grande variété de systèmes d'exploitation (Solaris, Linux, Windows 2000, HP UX, etc.) et d'offrir une bonne fiabilité pour un faible coût de développement (186 lignes de Perl pour les OS Unix).

Le développement d'outils permettant la mise à jour de l'annuaire et intégrant les spécificités de la gestion des logins en enseignement (création en masse). Ce développement réalisé en Java a constitué le plus gros travail du projet.



2.2 Conclusion

Après une année de test en demi grandeur (4000 logins) durant l'année 99/00, nous sommes passés à un déploiement sur l'ensemble de l'établissement dès la rentrée 2000 (14 000 logins).
Le résultat de l'exploitation s'est révélé pleinement satisfaisant, la fiabilité est bonne (aucun problème en 2 ans) et surtout la gestion collaborative des logins a permis de faciliter les travaux pratiques transversaux entre composantes. En particulier, cette solution permet d'éviter la création de plusieurs comptes par utilisateur, comme c'était le cas avant l'existence de l'annuaire.

Toutefois ce projet fait apparaître 2 limites :

- Un problème de performances est apparu lorsque les listes d'accès (ACL) ont cru en complexité et en nombre. Nous discutons plus loin de l'impact des ACL sur les performances des serveurs LDAP.
- Un autre problème est apparu quand nous avons étudié l'intégration du serveur LDAP aux autres Systèmes d'Information (SI) de l'établissement : comment faire dialoguer notre annuaire avec les applications de gestion de scolarité (APOGEE) ? Que faire quand un étudiant commence ses travaux pratiques avant d'être inscrit ?…

3. Le projet AGALAN

3.1 AGALAN regroupe un consortium d'universités :

Université Joseph Fourier – Grenoble 1 (Responsable du Projet)
Université Pierre Mendès-France – Grenoble 2
Université Stendhal - Grenoble 3
Université de Savoie
Institut National Polytechnique de Grenoble
IUFM de Grenoble
Université Louis Lumière – Lyon 2

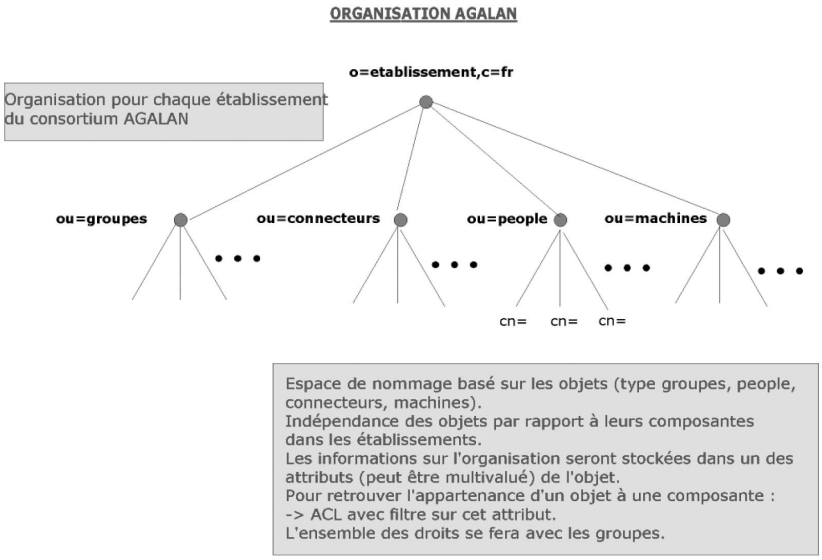
3.2 Les objectifs du projet AGALAN :

1. Construire un annuaire LDAP ayant une structure commune pour l'ensemble des établissements du consortium AGALAN, permettant la mise en place d'une authentification unique à l'ensemble des établissements du consortium.

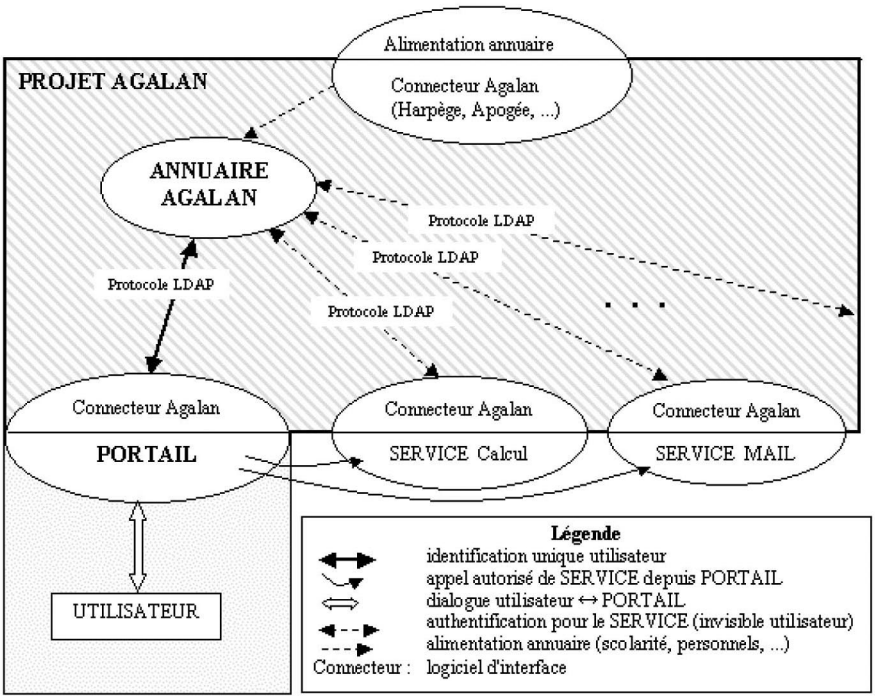
2. Réaliser et intégrer cet annuaire au cœur des différents systèmes d'informations et services dans les établissements. Cette phase implique notamment la réalisation des connecteurs avec les services existants.

3. Utiliser l'ensemble des connecteurs comme la ressource fondamentale utilisée par le futur portail Universitaire (GRECO, Lyon 2) pour l'accès authentifié à l'ensemble des services associés.

4. Elaborer les connecteurs en s'appuyant sur des méthodes de travail coopératif. Chaque connecteur sera développé une seule fois pour l'ensemble des établissements. Plusieurs équipes seront actives simultanément pour raccourcir les délais de réalisation. Ces équipes pourront être universitaires et/ou extérieures en fonction des moyens, des compétences et des partenariats.



3.3 Architecture du projet



4. Annuaires / SGBD

Cette partie discute des différents concepts des annuaires LDAP et des SGBD (Systèmes de Gestion de Bases de Données). Une comparaison de ces 2 technologies est présentée dans cette section.

4.1 Annuaires LDAP

Historiquement issus de la normalisation réseau X500, les annuaires LDAP offrent de très bonnes performances en consultation. Les accès en lecture ont été privilégiés au détriment du maintien de la cohérence de l'information. Ils ont adopté une structure de données arborescente bien adaptée à la description des organisations hiérarchiques mais qui ne permet pas de modéliser des informations complexes.

4.2 SGBD :

Un SGBD regroupe un ensemble d'outils pour la modélisation et la manipulation de données persistantes, le contrôle des accès concurrents (transactions) et la gestion de la sécurité et de la confidentialité. Cette technologie est aujourd'hui mature mais les SGBD n'offrant aucun protocole réseau standard restent inadaptés à la communication avec des équipements hétérogènes.

4.3 Comparaison des modèles de données

LDAP : le modèle de données LDAP est de type hiérarchique. Les données sont structurées en arbre. La racine de l'arbre est le point d'entrée dans l'annuaire. Le langage de manipulation de données, de type navigationnel, utilisé par ce modèle impose de spécifier le chemin d'accès aux données en naviguant dans le graphe de l'annuaire.
Il n'y a donc pas d'indépendance entre les programmes qui accèdent aux données et la structure interne des données (l'arborescence). L'inconvénient majeur est que toute modification dans la structure des données doit être répercutée sur les programmes.
De plus, la représentation hiérarchique des données interdit tout partage de l'information : pour modéliser le fait qu'une personne intervient dans plusieurs composantes (avec toujours les mêmes informations associées), il faut lui associer, dans l'annuaire, autant d'entrées que de composantes auxquelles elle appartient. Cette duplication de l'information alourdit la gestion des droits des utilisateurs et multiplie les risques d'erreurs (lors de la modification d'une information associée à une personne, il faut penser à modifier cette information à tous les endroits où elle apparaît).

SGBD : les SGBD offrent un modèle logique des données (classiquement le modèle relationnel) indépendant de leur représentation physique. Les programmes accèdent aux données via le modèle logique. Cette indépendance données-programmes permet de modifier la structure physique des données (par exemple pour optimiser les accès) sans toucher aux programmes d'application. Le modèle relationnel de données est le plus couramment utilisé, il permet de décrire simplement les données sous forme de tables (relations). Des langages déclaratifs (faciles d'utilisation) de définitions et de manipulations de données sont proposés. Le plus connu est SQL (Structured Query Language) qui fait l'objet d'un standard depuis 1989.
De plus, des contraintes d'intégrités (clef primaire, clef étrangère, restriction de domaine, etc.) peuvent être définies dans le schéma de la base de données. Leur maintien est automatiquement assuré par le SGBD, ce qui décharge d'autant les programmes d'applications.

4.4 Protocoles réseaux :

Au niveau des SGBD il n'existe pas vraiment de protocole réseau normalisé. Pour l'interfaçage entre les couches réseaux (TCP/IP, IPX…) et le SGBD les différents éditeurs proposent des solutions le plus souvent propriétaires. Oracle fournit par exemple une couche réseau NET 8 au dessus de TCP/IP ou un autre protocole réseau pour accéder à ses bases de données.

Le langage SQL est certes un standard pour l'accès aux données, mais il se situe au niveau applicatif. Il ne peut donc pas être utilisé pour la communication au niveau des couches réseaux.

Il existe cependant des tentatives de standardisation telles que :

- ODBC (Open DataBase Connectivity, Microsoft) : ODBC permet la communication des applications avec n'importe quel SGBD à condition que celui-ci possède une interface ODBC. Sur le plan théorique ODBC offre une connexion à une base de données indépendamment du SGBD. Mais cette technologie reste une solution propriétaire Microsoft qui ne fonctionne que sur les plates-formes Microsoft Windows !
- JDBC (Java DataBase Connectivity) : comme ODBC, JDBC permet la communication à une base de données. Face au succès de Java, la plupart des éditeurs de SGBD proposent les interfaces JDBC associées, ce qui en fait un standard de fait. Cette solution n'est bien sûr envisageable que si les équipements sont en mesure d'exécuter des programmes Java.

4.5 Performances :

Il est difficile de comparer les performances d'un annuaire et d'une base de données. D'une part il faut tenir compte des différentes implémentations par les différents constructeurs aussi bien pour les annuaires que pour les bases de données. La performance est d'abord liée à la qualité du logiciel. Il existe suffisamment de disparité dans la milieu des BDs (ORACLE vs MySQL, Access, SQLServer..) et dans celui des annuaires cf 6.4 comparatifs) pour ne pas tenir compte de cet aspect.

En terme uniquement de performance un annuaire sera plus rapide qu'un SGBD. Les SGBD sont ralentis par le fait qu'ils utilisent un modèle « coûteux » en temps : les jointures entre relations sont très pénalisantes en coût, la vérification permanente de la cohérence des informations (clés primaires et étrangères, triggers…) et l'utilisation d'un langage interprété (SQL) pour la manipulation des données sont autant de facteurs de ralentissement.

LDAP étant avant tout un protocole réseau et ne faisant pas de contrôle de cohérence il est forcément plus rapide.

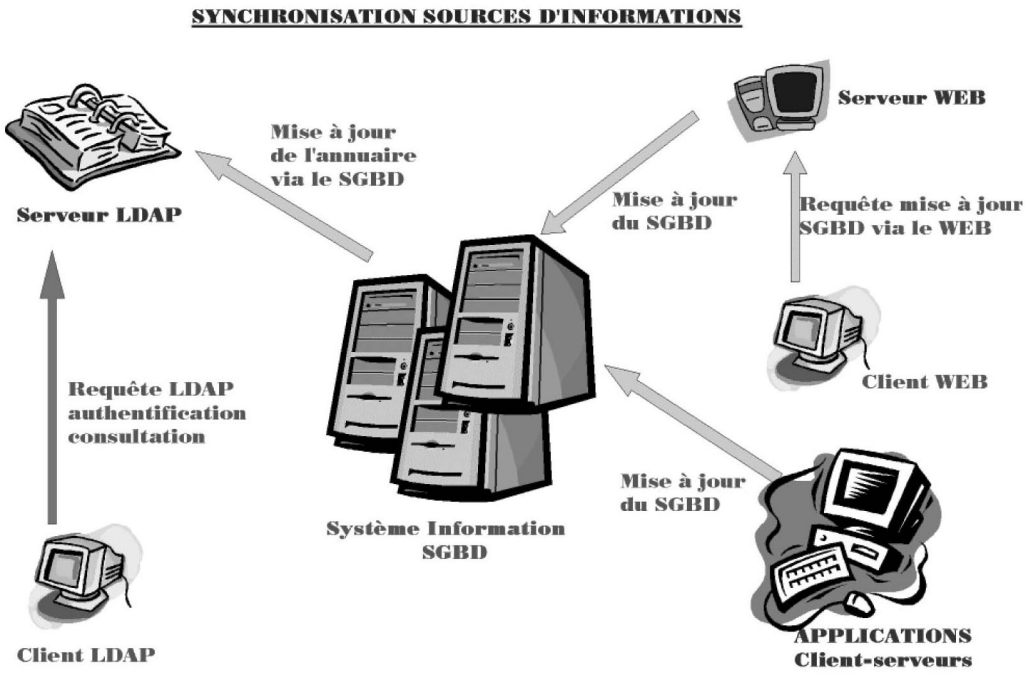
Mais d'une manière générale les questions que l'on peut se poser : la performance d'un annuaire ou d'une base de données peut-elle se limiter à ses temps d'accès et de réponses ? Peut-on toujours privilégier la rapidité, en particulier pour les mises à jour, au détriment de la cohérence des données ?

On peut donc voir LDAP comme un protocole réseau standard permettant un accès rapide aux données mais sans garantir la cohérence de son contenu. Les bases de données se situant à l'opposé avec une forte priorité sur le maintien de l'intégrité des données mais ne bénéficiant pas d'un protocole d'accès standardisé et par conséquent moins rapide qu'un annuaire.

5. Quelques pistes vers des solutions

5.1 Synchronisation des sources d'information
5.1.1 Cohabitation et complémentarité LDAP - SGBD.

L'idée est d'intégrer ces 2 outils dans une solution finale : le SGBD pour la gestion de l'information et LDAP comme protocole réseau d'interrogation pour la partie authentification. LDAP devient alors une des interfaces du système d'information pour l'accès aux données. La mise à jour de l'annuaire se fait via le(s) SGBD(s) uniquement.



5.1.2 Extraction SGBD -> LDAP

Faire une extraction automatique des données d'un SGBD vers un annuaire LDAP est une chose relativement aisée. Le format de données LDIF (LDAP Data Interchange Format) permet d'alimenter simplement un annuaire avec des scripts.

A l'usage 2 problèmes apparaissent :

- Ce mode de synchronisation est unidirectionnel : il présage que les données de l'annuaire ne seront jamais modifiées par le protocole LDAP. Cette limitation qui peut être acceptable dans le cadre d'un annuaire téléphonique peut devenir très limitative avec la multiplication des applications « directory enable ».
- Les objets sont ajoutés dans l'annuaire sans soucis de règles d'accès, il convient après l'importation de s'assurer de leur validité.

5.2 Passerelle SGBD/LDAP
5.2.1 Principe

Une passerelle SGBD/LDAP permet de représenter les données de l'annuaire (objects) dans une base de données (relations). OpenLdap offre, par exemple, la possibilité de mettre en « backend » du serveur LDAP une vraie base de données permettant d'utiliser SQL. Cependant ce passage se heurte à de nombreuses difficultés dues en particulier à la disparité des 2 modèles : hiérarchique et relationnel. Voici la liste non exhaustive des problèmes qui se posent :

- La notion d'attribut multivalué n'existe pas dans le modèle relationnel. Elle est même contraire à la norme (1° Forme Normale : atomicité des attributs). Donc si on veut représenter un objet avec une relation on ne pourra mettre que les attributs monovalués dans cette relation. Pour représenter, en relationnel, un objet ayant des attributs multivalués il faudrait autant de relations qu'il y a de possibilités de valeur dans chaque attribut multivalué… et donc autant de résultats possibles pour les requêtes.

- D'autre part les classes d'objets forment une hiérarchie de classes. Chaque objet hérite des attributs de la classe supérieure. Une entrée peut donc contenir des attributs de plusieurs objectclass.
  Il y a alors plusieurs possibilités :

  • soit faire une relation par classe d'objets et choisir une clé pour refaire la jointure et ainsi reconstruire l'objet.

  • soit ne faire qu'une relation en y mettant l'ensemble des attributs (ceux de la classe d'objets et ceux des classes supérieures).

- Enfin avec le modèle hiérarchique de LDAP une requête comme (cn=*), avec search scope égal à subtree, spécifiant donc une recherche en profondeur de tous les cn dans le sous-arbre est relativement simple. Par contre comment faire la même recherche en relationnel ? Faire une requête sur toutes les relations de la base ? Ce qui en terme de performance est impensable. Il faut donc reconstituer la notion de hiérarchie dans la base de données en rajoutant une ou des relations pour faire apparaître la notion arborescente traduite à travers le distinguished name (DN).

Le passage d'un modèle hiérarchique au modèle relationnel (et inversement) est donc un problème très complexe. Il faut « éclater » les différents objets de l'annuaire pour les stocker dans de multiples relations de la base de données. Cela implique à chaque requête de reconstruire les objets en faisant des jointures sur ces relations pour consulter ces données dans l'annuaire. Tout ceci est énormément pénalisant sur les performances.

5.2..2 Oracle Internet Directory

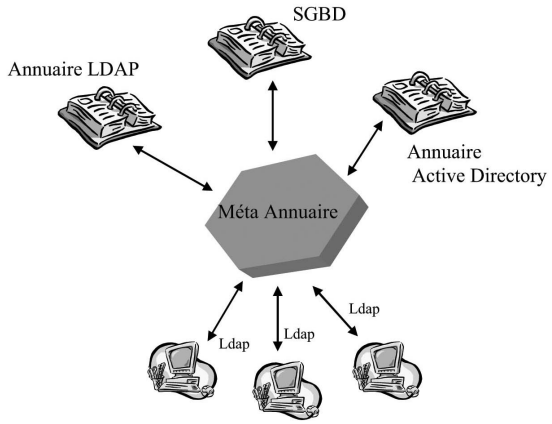
Pour illustrer cette complémentarité entre annuaires et SGBD nous avons testé Oracle Internet Directory, l'annuaire LDAP d'ORACLE, pensant qu'une solution complémentaire pouvait venir d'un spécialiste des bases de données.

Le choix qu'a fait Oracle est de se positionner sur le marché actuel des serveurs LDAP mais sans cependant exploiter la puissance de son modèle de données relationnel. L'annuaire LDAP utilise bien une base de données pour le stockage des informations (données, schéma, ACL) mais ne permet aucun contrôle de cohérence des données (aucune contrainte d'intégrité n'est définie dans le schéma de l'annuaire). D'autre part, il ne permet pas de synchroniser de manière bi-directionnelle l'annuaire et la base. Ce choix s'explique par le fait que les contrôles de cohérences nuisent à l'efficacité du serveur LDAP.

Oracle Internet Directory implémente une passerelle entre le modèle hiérarchique et le modèle relationnel (cf 5.2.1) pour le stockage des données et inversement pour la restitution des données. Au vue des faibles performances du produit on peut en déduire que soit l'implémentation de cette passerelle n'est pas optimisé, soit malgré toutes les optimisations possibles cette solution technique ne peut être satisfaisante en terme de rapidité.

5.3 Méta-Directory

Un méta-annuaire permet une vue unifiée d'un ensemble de sources d'informations de provenances variées (LDAP, SGBD, Active Directory, etc.). Ce type de logiciel se présente comme un annuaire LDAP. A la réception d'une requête, grâce à des connecteurs paramétrables, il interroge des sources diverses d'informations. La fusion de tout ou partie des réponses de ces connecteurs lui permettra de construire une réponse à l'interrogation via le protocole LDAP.



Les méta-annuaires seraient-ils la solution idéale ? Malheureusement deux problèmes de poids viennent relativiser notre enthousiasme :

- le prix,
- la complexité de configuration.

Microsoft fournit son « MMS (Microsoft Metadirectory Service) » mais avec l'obligation pour le client de recourir à une société de service agréée pour son installation et sa configuration.

Le logiciel « Meta-Directory 1.0 » de Iplanet (ex produit Netscape) était commercialisé en 99 à un prix de 15 US$ par utilisateur ce qui pour un établissement universitaire conduit rapidement à une facture d'un montant prohibitif.

5.4 Proposition de découpage de l'information

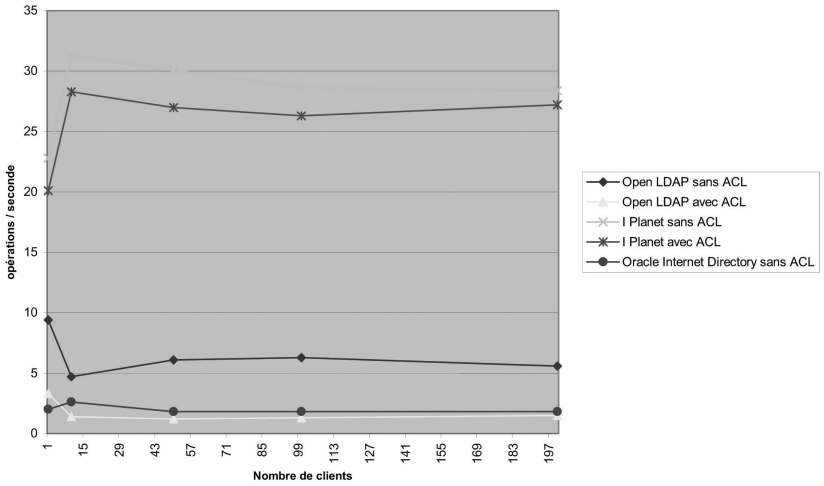
Comment alors parvenir à faire dialoguer un outil aussi « rustique » qu'un annuaire avec un SGBD actuel ?
Si l'offre logicielle n'apporte pas de solution, quelques règles simples permettent de minimiser les problèmes.

- Minimiser le nombre des informations communes au SGBD et à l'annuaire. Lors de la mise en place d'un annuaire, la tentation est grande de multiplier les informations qui y sont stockées poussé par « ça peut toujours servir ». Il faut garder en mémoire que toute information en double nécessitera la mise en place d'une méthode de synchronisation pour maintenir la cohérence des données. Avant d'enrichir son annuaire il faut toujours s'interroger sur l'utilité réelle des informations que l'on désire ajouter et leur rôle dans l'annuaire.
- Eviter de multiplier les « arborescences ».Il est tentant de multiplier les branches de l'arbre afin de faire correspondre la structure de nos annuaires à la complexité de nos organisations. Multiplier les branches permet aussi de simplifier les ACL en rattachant les autorisations aux branches. Pourtant la multiplication des branches rend très difficile le maintient de la cohérence de l'information dès qu'un même objet nécessite d'être référencé dans 2 branches différentes.

Dans le cadre du projet AGALAN le choix a été fait de garantir la cohérence des informations en évitant autant que possible la duplication des objets.

5.5 Performance et modélisation

Pour connaître l'impact de ces choix de modélisation sur les performances, il nous est apparu nécessaire de connaître l'impact des ACL sur les performances, nous avons testé trois logiciels serveurs LDAP (Open LDAP, Iplanet Directory Server, Oracle Internet Directory). Ces trois logiciels ont été soumis à une montée en charge simulée par un nombre croissant de clients (1,10,50,100,200). Les logiciels ont d'abord été testés sans ACL puis avec 190 lignes d'ACL (du type : le dn <uid=payan, o=ujf-grenoble, c=fr> a les droits en écriture sur les membres du groupe 10030)
Les résultats des mesures apparaissent dans le graphique ci-après :



Les résultats sont clairs : la qualité de l'implémentation des ACLs est fortement dépendante du logiciel serveur. Le choix d'une topologie « à plat » où toutes les informations de même type sont stockées au même endroit conduit à passer des ACL contrôlant l'accès par branche à des ACL contrôlant l'accès par groupe d'utilisateur et donc à une forte augmentation du nombre d'ACL.

Le choix de cette organisation de l'information nécessite un choix attentif du logiciel serveur si l'on veut éviter un effondrement des performances.

6. Conclusion

Comment dans un environnement complexe, aux multiples sources d'informations, aux nombreux protocoles réseaux, parvenir à fournir à ses utilisateurs une vision unifiée de ce monde ?
Il n'existe malheureusement pas de réponse unique à cette interrogation mais plusieurs pistes qui devront être développées en fonction de votre environnement :

- Le méta annuaire pour des « grands sites »
- Une réflexion approfondie sur la modélisation pour simplifier les mises à jour

Quel que soit votre choix il est important de ne pas perdre de vue qu'un annuaire LDAP est un système d'information supplémentaire dans votre établissement et que sa bonne utilisation dépend de la bonne modélisation de votre information. Deux particularités viennent encore compliquer votre tâche dans le cas d'un annuaire LDAP :

- Le caractère très « rustique » voir « primitif » de son organisation
- Le manque de maturité technique des logiciels.

7. Annexe : description des tests de performances effectués

Les résultats d'évaluation de performances fournis dans cet article ont été réalisés dans les condition suivantes :

7.1 Logiciels utilisés

Serveur :
Open LDAP Version 2.0.11
iPlanet Directory Server 5.0
Oracle Internet Directory 2.1

Client :
DirectoryMark1.2.1 *(www.mindcraft.com)* a été utilisé pour simuler l'activité de 1 à 200 clients. Les requêtes simulées tentaient de reproduire l'activité d'un serveur LDAP utilisé pour de l'authentification. Elles contenaient :

- 80% de recherche exact sur l'UID
- 10% de recherche avec « wildcard » sur le cn
- 10% de recherche exact sur le cn

une connexion LDAP (bind) avec une identité aléatoire par requête.
Le jeu de données utilisé pour remplir les annuaires testés sont les données de l'annuaire actuellement en exploitation sur notre établissement soit 21118 entrées. Notre volonté était de reproduire le plus fidèlement possible les conditions rencontrées au sein de notre établissement (assez représentatif d'un établissement universitaire de taille moyenne en France) en utilisant des données réelles.

7.2 Matériel utilisé (identique pour les 3 logiciels testés)

Serveur	Client
Sun 220 R	Sun Entreprise 3500
2 processeurs Ultra Sparc II (450 Mhz)	4 processeurs Ultra Sparc (400 Mhz)
1 Go de mémoire	1 Go de mémoire
36 Go de disque (SCSI)	200 Go de disques (F-CAL)
système Solaris 8	Système Solaris 8
connexion réseau Ethernet 100b/s Full Duplex	connexion réseau Ethernet 100b/s Full Duplex

L'interconnexion réseau entre les 2 équipements et réalisé par un commutateur « Accelar » de Bay Network.

8. Références

- Fundamentals of Databases (Elmasri and Navathe 3rd edition, 2000, Addison-Wesley).
- Oracle : *http://www.oracle.com*
- Sun : *http://www.sun.com*
- OpenLdap.org : *http://www.openldap.org*
- LDAP V3 : *http://www.ietf.org/rfc/rfc2251.txt*
- DirectoryMark 1.2.1 *http://www.mindcraft.com*