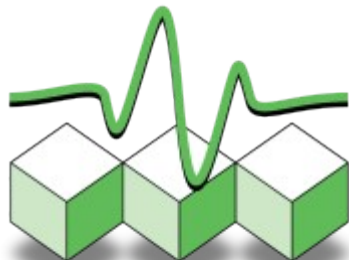


LINAGORA

Formation



Administration et sécurité



OpenLDAP™

<http://www.OpenLDAP.org>

Le logiciel OpenLDAP

Auteurs :

- Clément OUDOT, Raphaël OUAZANA et Sébastien BAHLOUL
- LINAGORA *Formation* : formation@linagora.com




Licence

Paternité - Pas d'Utilisation Commerciale - Partage des Conditions Initiales à l'Identique 2.0 France

Vous êtes libres :

- de reproduire, distribuer et communiquer cette création au public,
- de modifier cette création.

Selon les conditions suivantes :

-  Paternité. Vous devez citer le nom de l'auteur original.
-  Pas d'Utilisation Commerciale. Vous n'avez pas le droit d'utiliser cette création à des fins commerciales.
-  Partage des Conditions Initiales à l'Identique. Si vous modifiez, transformez ou adaptez cette création, vous n'avez le droit de distribuer la création qui en résulte que sous un contrat identique à celui-ci.

À chaque réutilisation ou distribution, vous devez faire apparaître clairement aux autres les conditions contractuelles de mise à disposition de cette création.

Chacune de ces conditions peut être levée si vous obtenez l'autorisation du titulaire des droits.

Ce qui précède n'affecte en rien vos droits en tant qu'utilisateur (exceptions au droit d'auteur : copies réservées à l'usage privé du copiste, courtes citations, parodie...)

Pourquoi LINAGORA met ce support sous licence Creative Commons

- Volonté de contribuer activement à l'essor du logiciel libre
- Promouvoir l'échange et favoriser l'émulation communautaire
- Assurer la pérennité de l'industrie logiciel libre et ne comptabiliser que la Valeur Ajoutée (le formateur)
- Partager le savoir et la connaissance à une vaste échelle

LINAGORA croit au Libre !

Présentation du formateur

- Parcours du formateur

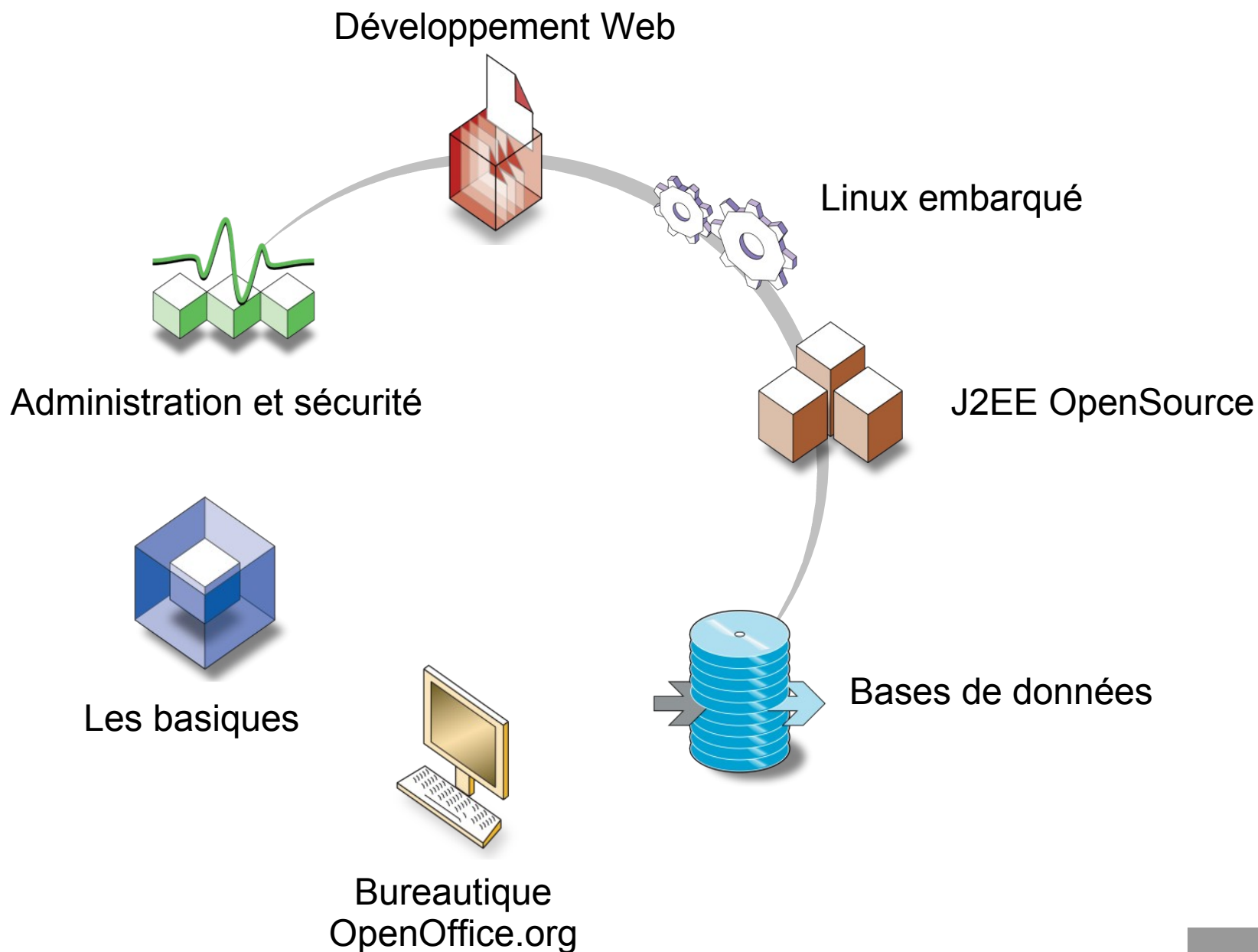
LINAGORA, premier EOS

- Créateur des concepts SS2L (Société de Services en Logiciels Libres) et TM2L (Tierce Maintenance Logiciel Libre), LINAGORA se définit désormais comme un Éditeur Orienté Service (EOS).
- LINAGORA propose une **Open Source Software Assurance** (OSSA) sur 150 logiciels libres :
 - Prêts à l'industrialisation, sur une plate-forme unique : le **08000LINUX.com**.
 - Avec garantie de service contractuelle : en cas de bug, LINAGORA s'engage au résultat sur des délais de résolution.
 - Gestion de la feuille de route du logiciel pour le compte du client et s'engage au reversement des développements.
- LINAGORA apporte également son expertise sur toute une gamme de **services professionnels** et de **formations** au travers de **LINAGORA Formation**.

LINAGORA Formation

- **7 années d'expérience**, au service des technologies libres et Open Source
- **40 modules** de formation répartis au travers de **7 filières**
- Un cadre agréable, au cœur de Paris
- Deux salles de formation climatisées pouvant accueillir jusqu'à 10 stagiaires.
- **2006 : Plus de 150 stages** effectués
- **2006 : Plus de 900 stagiaires**
- **Une satisfaction** moyenne client de **18/20**
- **Une note moyenne formateur** de **16,27/20**

Filières de formations



Organisation générale et planning

09h30 : début des cours

10h30 : pause du matin

10h45 : reprise des cours

12h00 : pause déjeuner

13h00 : reprise des cours

15h00 : pause de l'après-midi

15h15 : reprise des cours

17h30 : fin de journée

17h30 : libre discussion avec le formateur

Jour 1 :

- Présentation
- Installation
- Configuration
- Installation

Jour 2 :

- Sécurité
- Schéma
- Réplication
- Performances

Plan de cours

- Présentation du logiciel OpenLDAP
- Installation
- Configuration
- Utilisation (client et serveur)
- Sécurité
- Schéma
- Réplication
- Performances

Historique

- Issu du serveur LDAP de l'université du Michigan, dont dérive également Netscape Directory Server (devenu SUN Directory Server et Fedora Directory Server)
- Projet initié en 1998 (OpenLDAP v1), avec support LDAPv2
- Conforme LDAPv3 en 2000 (OpenLDAP v2)
- Version stable actuelle : OpenLDAP 2.3
- Version 2.4 en évaluation, stable courant 2007
- 3 développeurs principaux :
 - Howard Chu
 - Pierangelo Masarati
 - Kurt Zeilenga

Catégories de logiciels d'annuaires

- Annuaires systèmes :
 - Active Directory
 - Lotus
 - Novell eDirectory
- Annuaires généralistes :
 - OpenLDAP
 - ApacheDS
 - Sun ONE Directory Server
 - Fedora Directory Server
- Annuaires de sécurité :
 - Critical Path LiveContent (inJoin) Directory
 - IBM SecureWay

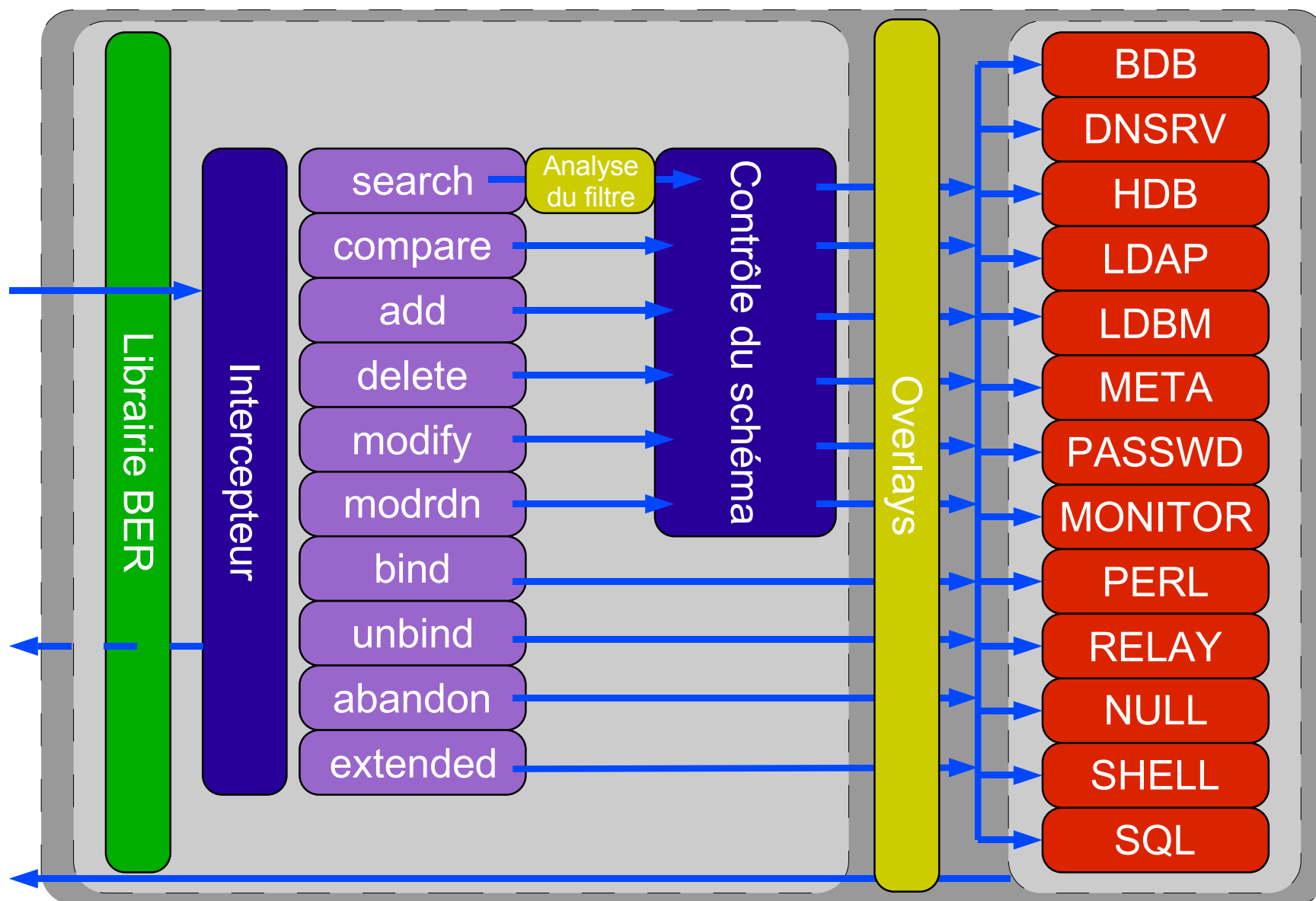
Fonctionnalités

- Serveur LDAP Open Source
- Contient :
 - Serveur indépendant (slapd) et serveur de réplication (slurpd)
 - Bibliothèques de connexion (libldap, liblber)
 - Commandes LDAP (ldapsearch, ldapadd, ldapmodify, ldapdelete, ...)
 - Commandes de gestion du contenu (slapadd, slapcat, slapindex, ...)
 - API (C, C++, TCL, Java)
- Supporte :
 - LDAPv2 et LDAPv3
 - Réplication complète et différentielle
 - Délégation d'authentification SASL / GSSAPI
 - Internationalisation UTF-8 via Unicode

Particularités

- Choix de backends :
 - BerkeleyDB (stockage)
 - LDAP et meta (mandataire)
 - Monitor (supervision)
 - SQL, Perl, Shell (langages de programmation)
- Choix d'overlays :
 - Politique des mots de passe
 - Groupes dynamiques
 - Intégrité référentielle
 - Réécriture des requêtes à la volée
- Configuration accessible par LDAP (branche cn=config)

Fonctionnement interne



Liste des backends

- back-ldbm :
 - Historiquement le premier backend de stockage
 - Basé sur une API DBM supportant gdbm (base de donnée GNU), ldbm (BerkeleyDB) et mdbm
 - Support de toutes les fonctionnalités historiques : alias, referrals
 - Développement et maintenance stoppés
 - Supprimé des prochaines versions
- back-bdb :
 - Backend existant depuis la version 4 des BerkeleyDB
 - Backend transactionnel : une modification ne nécessite que le verrouillage d'une page de la base de données
 - Support des fonctionnalités récentes : referrals, VLV, groupes dynamiques ...
 - Backend stockant des historiques permettant de reconstruire la base en cas de crash.
 - Utilisation de slapcat, slapindex et slapadd à chaud

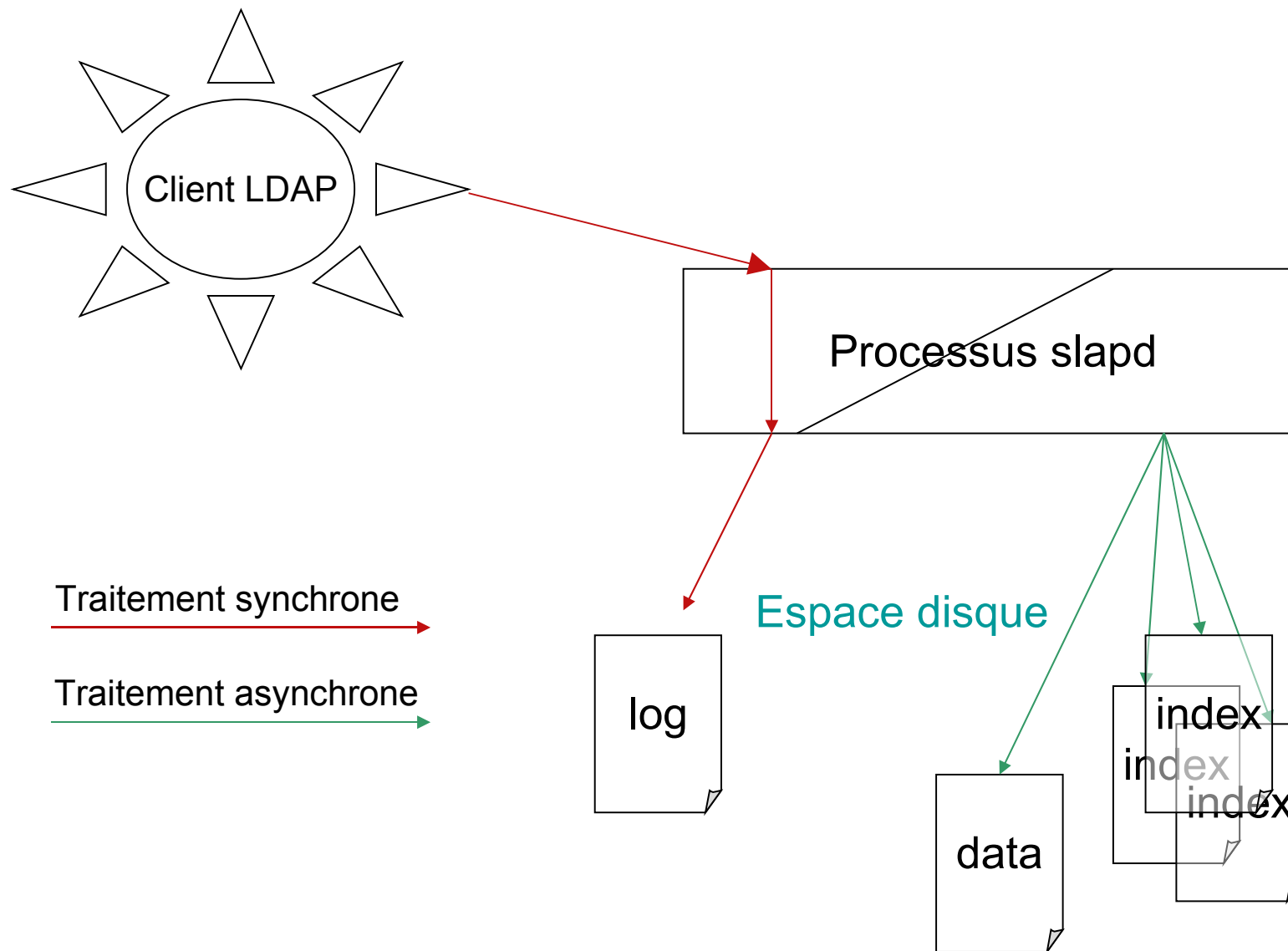
Liste des backends

- back-hdb :
 - Backend dérivé de back-bdb
 - optimisation du temps d'écriture
 - diminution de la redondance d'informations
 - réorganisation de l'information pour une optimisation de la consultation hiérarchique
- back-ldap :
 - Backend permettant la mise en place d'un mandataire (proxy)
 - Support de l'équivalence entre les attributs
 - Support de la réécriture entre des contextes de nommage différents
- back-monitor :
 - Ne contient pas de données
 - Est interrogé pour extraire des informations générales sur l'utilisation du serveur

Liste des backends

- back-meta :
 - Équivalent d'un back-ldap « multiple » : gère l'agrégation de plusieurs arbres en un unique
 - Ne gère pas l'agrégation d'entrées
 - Permet de constituer un méta annuaire dit de « virtualisation » simple (ne permet pas d'agréger des sources autres que des serveurs LDAP)
- back-shell / back-perl / back-tcl :
 - Backends permettant d'intercepter par des scripts shell / perl / tcl les différents appels aux méthodes LDAP
 - Extrêmement peu performants
 - Utilisés uniquement pour de la consolidation d'informations (interception des appels de changement de mots de passe, ...)
- back-sql :
 - Offre une interface LDAP aux bases de données
 - Nécessite de configurer l'association des tables et colonnes aux entrées et aux attributs

Fonctionnement du backend BDB



Enregistrement local des données

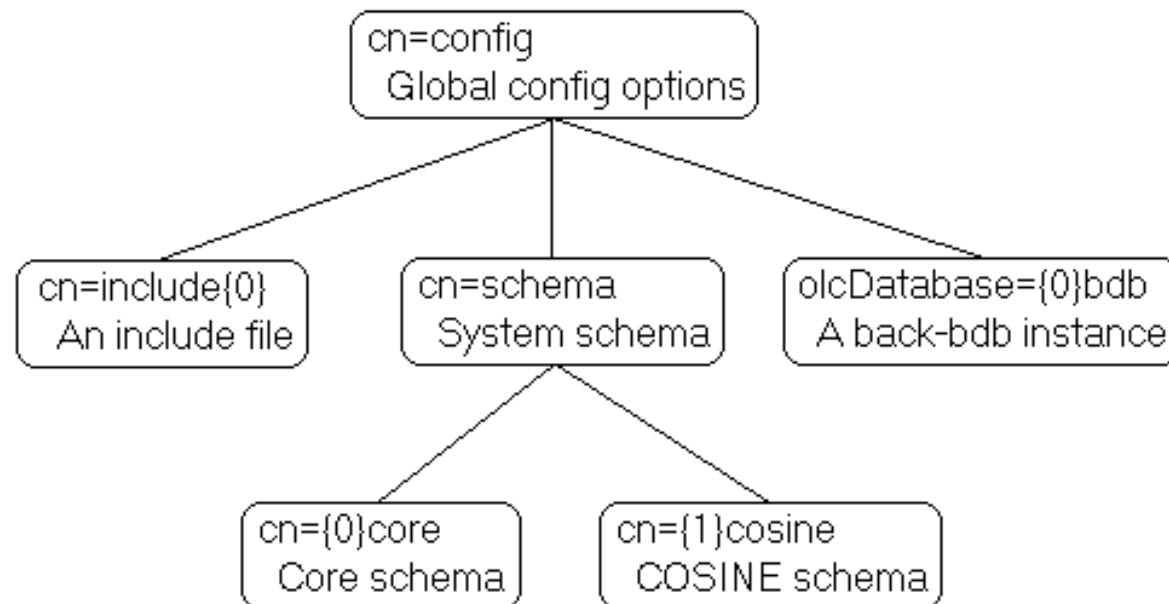
- *dn2entry.dbb* : équivalence entre les DN et les identifiants internes de stockage :
dc=linagora,dc=com : 0x000001
ou=personnes,dc=linagora,dc=com : 0x01094
uid=joe,ou=personnes,dc=linagora,dc=com : 0x41367
- *id2entries.dbb* : stockage des données de chaque entrée (fichier utilisé par la commande slapcat) :
0x41367 :
uid : joe
objectClass : top, person, inetOrgPerson, ...
sn : Joe
givenName : Jack
< ... />

Enregistrement local des données

- XYZT.dbb : fichier d'index sur l'attribut XYZT (quelque soit la forme d'indexage, fichier(s) utilisé(s) par la commande slapindex)
- Exemple de fichier sn.dbb pour un indexage de type eq et subfinal :
 - G: < ... />, 0x41367, < ... />
 - Ge: < ... />, 0x41367, < ... />
 - Ger: < ... />, 0x41367, < ... />
 - Germ: < ... />, 0x41367, < ... />
 - Germa: < ... />, 0x41367, < ... />
 - German: 0x41367

Le backend de configuration

- Fonctionnalité apparue en évaluation dans OpenLDAP 2.2, et en version stable dans OpenLDAP 2.3.
- Configuration accessible par le protocole LDAP, en interrogeant la branche cn=config :
 - `ldapsearch -x -D "cn=admin,dc=linagora,dc=com" -s base -b "cn=config"`



Le backend de configuration

World

- config
 - {0}config
 - {1}bdb
 - {-1}frontend
 - {2}monitor**
- include{0}
- schema
 - {0}core
 - {1}test
 - {2}test
 - {3}test

Vue HTML | Editeur de Table

attribute type	value
objectClass	olcDatabaseConfig
olcDatabase	{2}monitor
olcLastMod	TRUE
olcMaxDerefDepth	15
olcReadOnly	FALSE
olcAccess	
olcLimits	
olcPlugin	
olcReplica	
olcReplogFile	
olcRequires	
olcRestrict	
olcRootDN	
olcRootPW	
olcSchemaDN	
olcSecurity	
olcSizeLimit	
olcSubordinate	
olcSuffix	
olcSyncrepl	
olcTimeLimit	
olcUpdateDN	
olcUpdateRef	

Soumettre Ré-initialiser Changer une Classe Propriétés

Plan de cours

- Présentation du logiciel OpenLDAP
- Installation
- Configuration
- Utilisation (client et serveur)
- Sécurité
- Schéma
- Réplication
- Performances

Modes d'installation

- Depuis les paquets de la distribution Linux :
 - RedHat, Fedora Core et RHES :
 - Obligatoires : openldap-2.3.X.rpm (librairies obligatoires)
 - À installer : openldap-servers-2.3.X.rpm et openldap-clients-2.3.X.rpm
 - Debian, Ubuntu :
 - Obligatoires : libldap2
 - À installer : slapd, ldap-utils
 - Toutefois les choix de compilation sont imposés ce qui peut ne pas convenir à la maîtrise nécessaire à ce type de composant essentiel !
- Depuis les sources :
 - Besoin des outils de compilation C (autoconf, make, gcc)
 - Besoin des bibliothèques (fichiers .h) des produits tiers
 - Permet de sélectionner finement les fonctionnalités et d'optimiser les binaires

Pré-requis à l'installation

- Avant de lancer la configuration des sources, il faut déterminer :
 - Le format de stockage (backend)
 - Les fonctionnalités (overlays)
 - Le modèle de sécurité (SSL/TLS, SASL)
 - L'architecture des annuaires (classique ou multimaîtres)
- Et pré-installer les bibliothèques correspondantes :
 - SASL : cyrus-sasl
 - SSL/TLS : OpenSSL ou GnuTLS
 - Backend BDB, HDB : BerkeleyDB 4.2, 4.4 ou 4.5 (pas la 4.3 !)
 - Backend SQL : UnixODBC

Installation de BerkeleyDB par les sources

- Télécharger une version de BerkeleyDB depuis le site et les correctifs associés :
 - <http://www.oracle.com/technology/software/products/berkeley-db/index.html>
- Désarchiver :
 - `tar xzf db-4.x.y.tar.gz`
- Appliquer les correctifs :
 - `patch -p1 < patch.4.x.y.z`
- Détermination automatique de la configuration locale :
 - `cd db-4.x.y/build_unix`
 - `../dist/configure`
- Calcul des dépendances, compilation et installation :
 - `make depend && make`
 - `sudo make install`
- Le répertoire d'installation par défaut est :

/usr/local/BerkeleyDB.4.x

Installation d'OpenLDAP par les sources

- Télécharger la dernière version depuis le site :
 - `wget -c ftp://ftp.openldap.org/pub/OpenLDAP/openldap-release/openldap-2.3.35.tgz`
- Désarchiver :
 - `tar xzf openldap-2.3.35.tgz`
- Regarder les options disponibles :
 - `cd openldap-2.3.35/`
 - `./configure --help`
- Si besoin, indiquer l'emplacement des produits tiers :
 - `export CPPFLAGS="-I/usr/local/BerkeleyDB.4.x/include"`
 - `export LDFLAGS="-L/usr/local/BerkeleyDB.4.x/lib -R/usr/local/BerkeleyDB.4.x/lib"`
 - `export LD_LIBRARY_PATH=/usr/local/BerkeleyDB.4.x/lib`

Installation d'OpenLDAP par les sources

- Indiquer les options (dont le répertoire d'installation) :
 - `./configure --enable-bdb --with-tls --without-cyrus-sasl --prefix=/opt/openldap --enable-monitor --enable-overlays`
- Calcul des dépendances et compilation :
 - `make depend && make`
- Passage des tests (optionnel) :
 - `make test`
- Installation :
 - `sudo make install`

Exercice

- Installer les produits tiers à partir des paquets fournis dans votre distribution Linux :
 - BerkeleyDB 4.2
 - OpenSSL
- Installer OpenLDAP à partir des sources dans le répertoire utilisateur (par exemple /home/linagora/stage)

Plan de cours

- Présentation du logiciel OpenLDAP
- Installation
- Configuration
- Utilisation (client et serveur)
- Sécurité
- Schéma
- Réplication
- Performances

Organisation du répertoire d'installation

- bin : utilitaires clients
- etc : configuration
- include : fichiers d'en-têtes pour compiler des clients LDAP
- lib : bibliothèques pour compiler des clients LDAP
- libexec : exécutable des serveurs
- man : pages de manuel
- sbin : utilitaires d'administration
- var : données variables (bases, données de réplication, fichiers de lancement)

Fichiers de configuration

- Fichier principal de configuration du serveur :
 - `$prefix/etc/openldap/slapd.conf`
- Configuration (optimisation) de BerkeleyDB :
 - `$prefix/var/openldap-data/DB_CONFIG`
- Fichier de configuration des clients :
 - `$prefix/etc/openldap/ldap.conf`
- Localisation des fichiers schémas :
 - `$prefix/etc/openldap/schema/`
- Manuel intégré :
 - `man -M $prefix/man slapd.conf`

Structure de la configuration

- Structure de slapd.conf :
 - Directives générales :
 - Inclusion des fichiers schémas
 - Directives de sécurité
 - Contrôles d'accès (ACL)
 - Limites
 - Paramètres d'exécution
 - Définition d'un backend (backend XXX) :
 - Paramètres communs aux bases du backend
 - Définition d'une base (database XXX)
 - Paramètres spécifiques à la base (index, mots de passe, etc.)
 - Overlays
 - (Définition d'une autre base)
 - (Définition d'un nouveau backend)

Paramètres généraux

- *referral* : envoie des clients sur cette URL si le serveur est incapable de répondre
- *threads* <integer> : nombre de fils d'exécution maximum
- *tool-threads* <integer> : nombre de fils d'exécution maximum pour les utilitaires d'administration
- *pidfile* : fichier contenant l'identificateur de processus du serveur
- *argsfile* : fichier contenant les arguments passés au lancement du serveur
- *gentlehub* {on|off} : arrêt sans coupure des connexions établies
- *rootDSE* <filename> : nom du fichier LDIF contenant les informations à publier sur l'entrée dn=« » de l'annuaire

Paramètres généraux

- *reverse-lookup* {on|off} : active ou non la résolution de nom pour l'historisation (logs)
- *loglevel* : Niveau de debug enregistré par syslog
- *sockbuf_max_incoming* : taille maximum acceptée pour une session anonyme
- *sockbuf_max_incoming_auth* : taille maximum acceptée pour une session authentifiée
- *default_search_base* <dn> : base par défaut en l'absence de ce paramètre dans une requête
- *schemadn* <dn> : nom de l'attribut contenant le DN du schéma ajouté par le serveur sur chaque entrée

Paramètres globaux des backends

- *suffix* : Précision du contexte de nommage
- *lastmod* {on|off} : slapd maintient ou non les attributs opérationnels :
 - modifiersName
 - modifyTimeStamp
 - creatorsName
 - createTimeStamp
- *readonly* {on|off} : le contexte est ou non accessible en écriture
- *rootdn* : DN de l'administrateur du suffixe correspondant (ce champ doit faire partie du contexte de nommage auquel il est attaché)
- *rootpw* : mot de passe de l'administrateur (possibilité de le chiffrer avec slapasswd)

Paramètres spécifiques à BDB et HDB

- *cachesize* <integer> : Nombre d'entrées maintenues en mémoire (du processus OpenLDAP)
- *checkpoint* <kbytes> <min> : Point d'écriture du fichier de log lorsque soit il dépasse le volume indiqué par le nombre <kbytes>, soit <min> minutes sont passées
- *dbnosync* : pas de synchronisation instantanée sur disque (ni dans le fichier de log ni dans les fichiers de données)
- *dirtyread* : autorise la lecture d'une information non « commitée », c'est-à-dire uniquement présente dans le fichier de log
- *idlcachesize* <integer> : meilleure performance si *idlcachesize* = *cachesize* (pour HDB, *idlcachesize* = 3 * *cachesize*)
- *searchstack* <integer> : profondeur maximum d'une recherche (refus de filtres trop complexes pour éviter le déni de service)
- *index* <attrlist> [pres,eq,approx,sub,...] : ajoute un ou plusieurs index sur un ou plusieurs attributs

Note : penser à utiliser *slapindex* après l'ajout d'un index sur un attribut existant

Exercice

- Configuration basique d'OpenLDAP :
 - Inclure les schémas :
 - Core
 - Cosine
 - inetOrgPerson
 - Déclarer une base BDB :
 - Contexte : dc=linagora,dc=com
 - Administrateur : cn=admin,dc=linagora,dc=com (et choisir un mot de passe)
 - Indexer les attributs objectClass, cn et sn en équivalence et présence

Plan de cours

- Présentation du logiciel OpenLDAP
- Installation
- Configuration
- Utilisation (client et serveur)
- Sécurité
- Schéma
- Réplication
- Performances

Lancement du serveur

- slapd :
 - Lancement à la demande via inetd :
 - Pas de charge permanente pour un service peu utilisé
 - Peu utilisé, car slapd est « un service lourd »
 - Lancement unique sous forme de démon :
 - Disponibilité instantanée
 - Absence de relance d'un processus à chaque connexion
- slurpd :
 - Lancement en mode instantané (one-shot) :
 - Traite toutes les propagations à effectuer et s'arrête (utile pour resynchroniser sur une connexion non permanente)
 - Lancement en mode permanent (démon) :
 - Fonctionne en parallèle de slapd pour propager les modifications

Paramètres de lancement

- Options communes :
 - -d : niveau de debug (puissances de 2, de 2 à 2048, composables)
 - -f : fichier de configuration à utiliser
 - -l : processus syslog utilisé pour les logs (LOCAL0, LOCAL1, LOCAL4, ...)
- slapd :
 - -h : URL séparées par un espace (exemple : -h « ldap:/// ldaps:/// »)
 - -u, -g et -r : permettent de protéger le reste du système contre une défaillance du démon
- slurpd :
 - -r <relogfile>: fichier contenant les modifications notifiées par le daemon slapd
 - -t : répertoire temporaire dans lequel le fichier relogfile est copié avant d'être traité

Peuplement initial

- Utilisation d'un fichier LDIF pour insérer le DIT :
 - Contexte de nommage
 - Branches
 - Utilisateurs spéciaux (comptes applicatifs, réplication, etc.)
- Exemple de LDIF pour le contexte de nommage :

dn: dc=linagora,dc=com

objectClass: top

objectClass: dcObject

objectClass: organization

dc: linagora

o: LINAGORA

description: LINAGORA

Outils d'administration du serveur

- *slapacl* : teste les ACLs
- *slapauth* : teste les paramètres de connexion (authz)
- *slapcat* : extraction des informations des fichiers DB
- *slapadd* : ajout d'entrées sans passer par la couche réseau, mais directement par les librairies
- *slapindex* : recrée l'ensemble des fichiers d'index (peut être assez long !)
- *slapdn* : vérifie la conformité d'un DN en fonction du schéma du serveur
- *slaptest* : vérifie la conformité du fichier slapd.conf
- *slappasswd* : génère un mot de passe chiffré
- *slapindex* et *slapadd* ont une option -q améliorant grandement leurs performances

Clients LDAP en ligne

- *Idapsearch* : recherche dans l'annuaire
- *Idapadd* : ajoute une ou plusieurs entrées par le biais d'un fichier LDIF
- *Idapdelete* : supprime une ou plusieurs entrées, ou même un arbre
- *Idapcompare* : vérifie qu'une valeur existe ou non pour un attribut particulier d'une entrée
- *Idapmodify* : modifie des entrées dans l'annuaire
- *Idapmodrdn* : modifie le RDN et/ou déplace l'entrée
- *Idappasswd* : utilise l'opération étendue de mise à jour du mot de passe
- *Idapwhoami* : opération renvoyant l'identité de l'utilisateur

Options générales des clients LDAP

- -D <dn> : DN de l'entrée utilisée pour se connecter à l'annuaire
- -w <chaîne> : précise le mot de passe sur la ligne de commande
- -H <URL LDAP> : précise l'URL (hôte et port) de connexion à l'annuaire
- -h <hôte>
- -p <port>
- -P {2|3} : version du protocole LDAP utilisé pour la connexion
- -f : fichier LDIF contenant les modifications à apporter
- -c : mode continu, ne s'arrête pas aux erreurs
- -s : stocke les erreurs dans un fichier
- Options par défaut dans le fichier ldap.conf

Paramètres spécifiques à la recherche

- -b <base> : DN de départ de recherche
- -s <scope> : étendue de la recherche (base, one, sub)
- "filtre" : Filtre LDAP (exemple : "(objectClass=person)")
- Attributs :
 - Liste spécifique d'attributs (exemple : sn cn uid)
 - Tous les attributs ('*')
 - Tous les attributs opérationnels ('+')

Comparaison des méthodes

- Connecté (clients LDAP) :
 - Permet de prendre en compte les évolutions éventuelles durant le processus d'alimentation
 - Autorise la base LDAP à être disponible malgré l'alimentation
 - Utilise les mécanismes de réplication permettant d'assurer une synchronisation de tous les serveurs à partir d'un unique point d'accès
- Déconnecté (outils d'administration) :
 - Import rapide, sans transfert ni vérification réseau
 - Possibilité d'importer sans index pour réindexer par la suite (plus rapide)
 - Possibilité de limiter les vérifications pour accélérer le chargement (option -q)

Exercice

- Peuplement initial :
 - Créer un LDIF du contexte et d'une branche ou=personnes
 - Créer des utilisateurs dans cette branche avec des mots de passe
 - Insérer ce fichier avec slapadd
- Démarrer slapd en debug (niveau 256)
- Utiliser le client LDAP de recherche pour :
 - Voir la totalité des entrées
 - Voir les attributs opérationnels du contexte de nommage
 - Voir l'entrée RootDSE
- Utiliser les différents clients LDAP pour lire et éditer les données de l'annuaire

Plan de cours

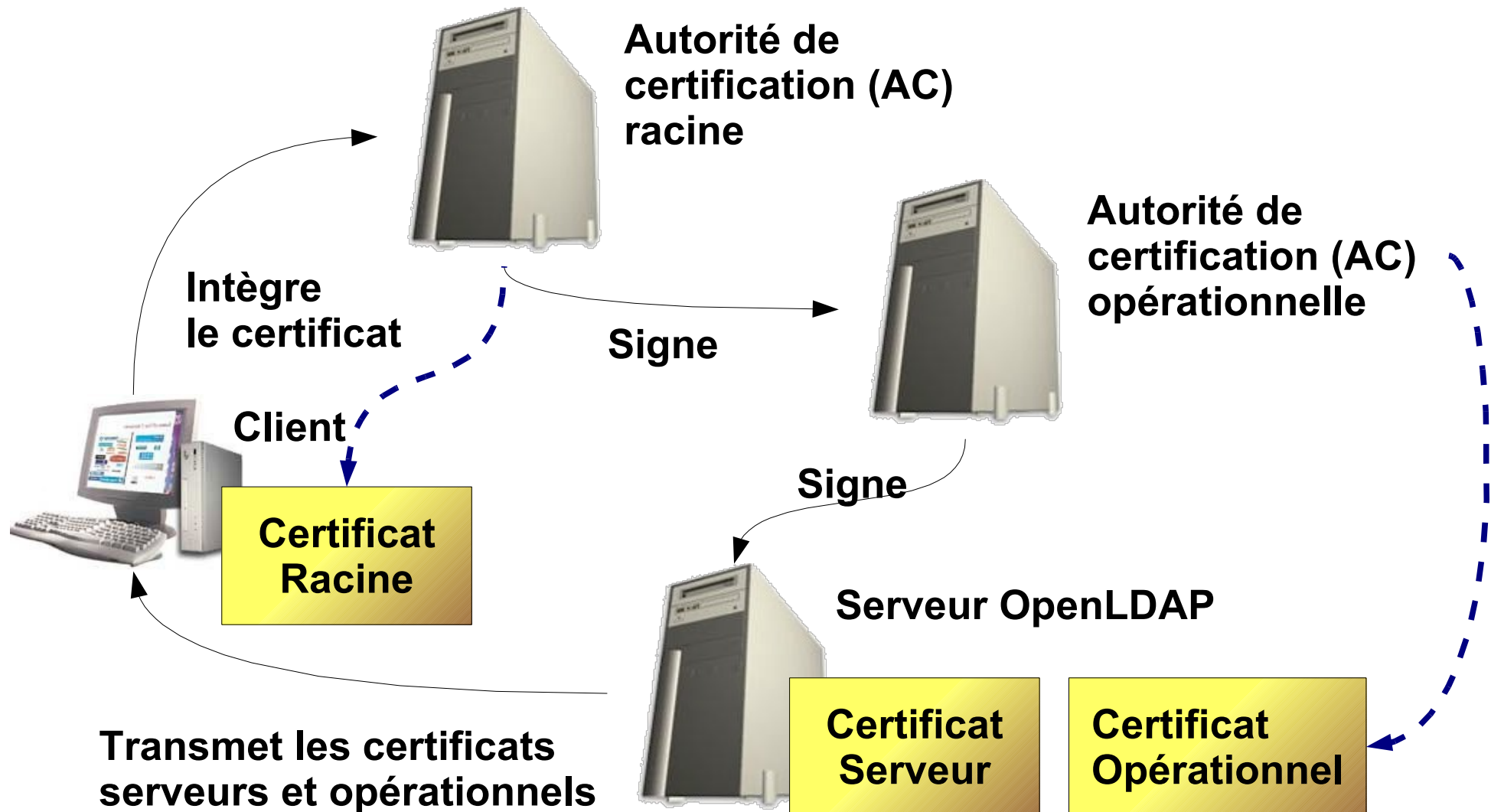
- Présentation du logiciel OpenLDAP
- Installation
- Configuration
- Utilisation (client et serveur)
- Sécurité
- Schéma
- Réplication
- Performances

SSL et TLS

- SSL : Secure Socket Layer (version actuelle : v3)
- TLS : Transport Layer Security (version actuelle : v1)
- Comparaison SSL/TLS :
 - mêmes algorithmes
 - SSL chiffre une connexion (tunnel)
 - TLS est inclus dans LDAP et est mis en place au cours de la connexion
 - SSL nécessite un port de connexion différent (636) , alors que TLS s'adapte sur le protocole LDAP (donc sur le port standard, 389) grâce à l'opération startTLS
- Pour les faire fonctionner il faut disposer d'un certificat valide :
 - signé d'une autorité internationale
 - signé d'une autorité locale reconnue par tous les clients

Flux SSL/TLS





Paramètres de sécurité dans slapd.conf

- *TLSCipherSuite* <ciphers>+ : permet de préciser quels sont les algorithmes de chiffrement utilisés, par exemple : HIGH:MEDIUM:+TLSv1
- *TLSCertificateFile* <filename> : fichier du certificat public du serveur
- *TLSCertificateKeyFile* <filename> : fichier contenant la clé privée du certificat du serveur
- *allow / disallow / require* <feature> : autorise / refuse / nécessite des connexions suivant certaines fonctionnalités (exemple : disallow LDAPv2)
- *password-hash* <hash> [<hash>...] : algorithme de cryptage utilisé lors de l'utilisation de l'opération étendue de changement du mot de passe

Paramètres de sécurité dans slapd.conf

- Facteurs de sécurité :
 - *security* <factors> : spécifie des longueurs de clés minimum à utiliser pour assurer les opérations :
 - *ssf*=<n>, *tls*=<n>
 - *update_ssf*=<n>, *update_tls*=<n>
- Réplication sécurisée :
 - Étant donné qu'en s'adressant à un serveur esclave, on ne peut pas savoir si la connexion a pu être sécurisée par TLS (même port de connexion), il est possible de forcer la réplication à ne s'effectuer que si la communication est chiffrée :
 - Dans la section *replica* du contexte de nommage, ajouter *tls=critical*
 - Dans la section *syncrepl*, ajouter *tls=critical*

Génération d'un certificat OpenSSL

- Génération de la clé privée :
 - `openssl genrsa -aes256 -out ca.key 1024`
- Création de la demande de signature :
 - `openssl req -new -key ca.key -out ca.csr`
- « Auto signature » de la demande :
 - `openssl x509 -days 1460 -signkey ca.key -in ca.csr -req -out ca.pem`
- Intégration du certificat dans OpenLDAP (slapd.conf) :
 - `TLSCipherSuite HIGH:MEDIUM:+TLSv1`
 - `TLSertificateFile $prefix/etc/openldap/ca.pem`
 - `TLSertificateKeyFile $prefix/etc/openldap/ca.key`
- Relancer OpenLDAP (slapd) avec les URL correspondantes :
 - `$prefix/libexec/slapd -h "ldap:// ldaps://"`

Règles de contrôles d'accès (ACL)

- Permet de déterminer quels sont les accès donnés à un client de l'annuaire
- Peut être déterminé par de nombreux critères
- Coût non négligeable d'évaluation. Peut donc nécessiter des spécialisations fonctionnelles dans certains cas.
- Paramètres :
 - access to <quoi> [by <qui> <droits> <contrôles>]+
- Manuel :
 - man -M \$prefix/man slapd.access

L'objet de l'ACL

- Sur quoi porte l'ACL : 3 éléments
 - Élément n°1 : quel DN
 - DN (expression régulière)
 - base : le DN lui - même
 - one : les fils directs du DN précisé
 - subtree : toute la descendance y compris le DN
 - children : toute la descendance sans le DN
 - Élément n°2 : sélection par rapport aux attributs
 - Filtre sur les attributs (comme ceux de ldapsearch)
 - Élément n°3 : quels « attributs » de ce(s) DN
 - Liste d'attributs (exemple : userPassword, uid)
 - L'entrée tout entière (entry)
 - Les fils de cette entrée (children)

Les droits d'une ACL

- Droits incrémentaux : none < auth < compare < search < read < write
 - auth = authentification (bind uniquement)
 - compare = opération de comparaison (compare)
 - search = permet d'utiliser l'attribut dans un filtre
 - read = lecture des valeurs de l'attribut
 - write = écriture (modification, ajout et suppression) des valeurs de l'attribut
- Droits séparés : {x,c,s,r,w}+
- Exemple pour l'attribut userPassword :

access to attrs=userPassword

by self xw+

by * x+

À qui s'applique l'ACL

- *** : tout le monde
- *anonymous* : toute connexion non authentifiée sur l'annuaire
- *users* : toute connexion authentifiée sur l'annuaire
- *self* : l'entrée utilisée pour s'authentifier
- *dn* : un DN spécifié
- *group* : un groupe d'utilisateur (groupOfNames par défaut)
- *peername, sockname* : en fonction des connexions (nom d'hôte, ...)
- *set* : définition en fonction des attributs des entrées
- *aci* : par les ACL stockées dans l'annuaire

Exemples d'ACL

- Lecture sur la branche des personnes :

access to dn.subtree="ou=personnes,dc=linagora,dc=com"

filter="objectClass=person"

attrs=uid,sn,givenName,mail,objectClass

by users csr+

by * none

- Protection du mot de passe :

access to attrs=userPassword

by self xw+

by uid=replicator,dc=linagora,dc=com write

by * x+

Les ACI

- Afin de permettre une plus grande dynamique des ACL, il est maintenant possible d'insérer des ACI dans l'annuaire.
- Nom de l'attribut : OpenLDAPaci
- Syntaxe :
 - OID # SCOPE # RIGHTS # TYPE # SUBJECT
 - OID : ordre de la règle (1, ..., n)
 - SCOPE : « entry »

Les ACI

- RIGHTS = ACTION [; PERMISSIONS ; TARGET] +
- ACTION = « grant » | « deny »
- PERMISSIONS = PERMISSION + [',' + PERMISSION] *
- PERMISSION = 'w' | 'r' | 's' | 'c' | 'a'
- TARGET = « all » | « children » | attribut
- TYPE = « access-id » | « group » | « self »
- SUBJECT = DN

Limites

- *limits* <who> <limit> [<limit>...]
 - Avec <who> :
 - anonymous
 - users
 - dn[.<style>]=]<pattern>
 - group[/oc[/at]]=<pattern>
 - Et <limit> :
 - time[.{soft|hard}]=<integer>
 - size[.{soft|hard|unchecked}]=<integer>
- Cas particuliers :
 - *sizelimit* : taille maximum d'une recherche
 - *timelimit* : temps maximum d'une recherche
 - *idletimeout* : temps maximum d'une connexion devenue inactive

Exercice

- SSL/TLS :
 - Créer et signer un certificat serveur
 - Installer le certificat dans OpenLDAP
 - Contrôler la sécurité avec les outils OpenSSL
- ACL et limites :
 - Inscrire des règles d'accès et des limites dans slapd.conf
 - Trouver des méthodes pour vérifier l'efficacité des règles mises en place

Plan de cours

- Présentation du logiciel OpenLDAP
- Installation
- Configuration
- Utilisation (client et serveur)
- Sécurité
- Schéma
- Réplication
- Performances

Les schémas

- Les schémas courants :
 - Core : l'ensemble des objets dont OpenLDAP a besoin
 - Cosine (RFC 1274) : issu des schémas X500
 - NIS : les informations pour les comptes informatiques
 - inetOrgPerson (RFC 2798) : une personne au sein d'une organisation
- Les schémas en cours :
 - Misc : objet de routage mail (abandonné)
 - Samba : Accès aux ressources sur un serveur Microsoft
 - Bind : support des objets DNS
 - Dyngroup : support des groupes dynamiques
- Visualisation du schéma d'un serveur :
 - `ldapsearch -b cn=subschema -x -s base '+'`

Exemples

- Validité des valeurs des attributs en fonction de la syntaxe :

attributetype (0.9.2342.19200300.100.1.20

DESC 'RFC1274: home telephone number'

NAME ('homePhone' 'homeTelephoneNumber')

EQUALITY telephoneNumberMatch

SUBSTR telephoneNumberSubstringsMatch

SYNTAX 1.3.6.1.4.1.1466.115.121.1.50)

- Syntaxe et normalisation associées :

IdapSyntax (1.3.6.1.4.1.1466.115.121.1.50

DESC 'Telephone Number')

Exemples

- Cohérence du contenu des entrées en fonction des classes d'objet :

objectClass (2.16.840.1.113730.3.2.2

NAME 'inetOrgPerson'

DESC 'RFC2798: Internet Organizational Person'

SUP organizationalPerson STRUCTURAL

MAY (audio \$ businessCategory \$ carLicense \$ departmentNumber \$ displayName \$ employeeNumber \$ employeeType \$ givenName \$ homePhone \$ homePostalAddress \$ initials \$ jpegPhoto \$ labeledURI \$ mail \$ manager \$ mobile \$ o \$ pager \$ photo \$ roomNumber \$ secretary \$ uid \$ userCertificate \$ x500uniqueIdentifier \$ preferredLanguage \$ userSMIMECertificate \$ userPKCS12))

- Règles d'égalité (d'approximation, de comparaison, ...)

matchingRules (1.3.6.1.4.1.1466.109.114.2

NAME 'caseIgnoreIA5Match'

SYNTAX 1.3.6.1.4.1.1466.115.121.1.26)

Exercice

- Schéma :
 - Définir deux attributs
 - Définir une classe d'objet dérivant de la classe inetOrgPerson possédant ces deux attributs de manière facultative
 - Installer le schéma dans OpenLDAP
 - Insérer des entrées conformes au nouveau schéma

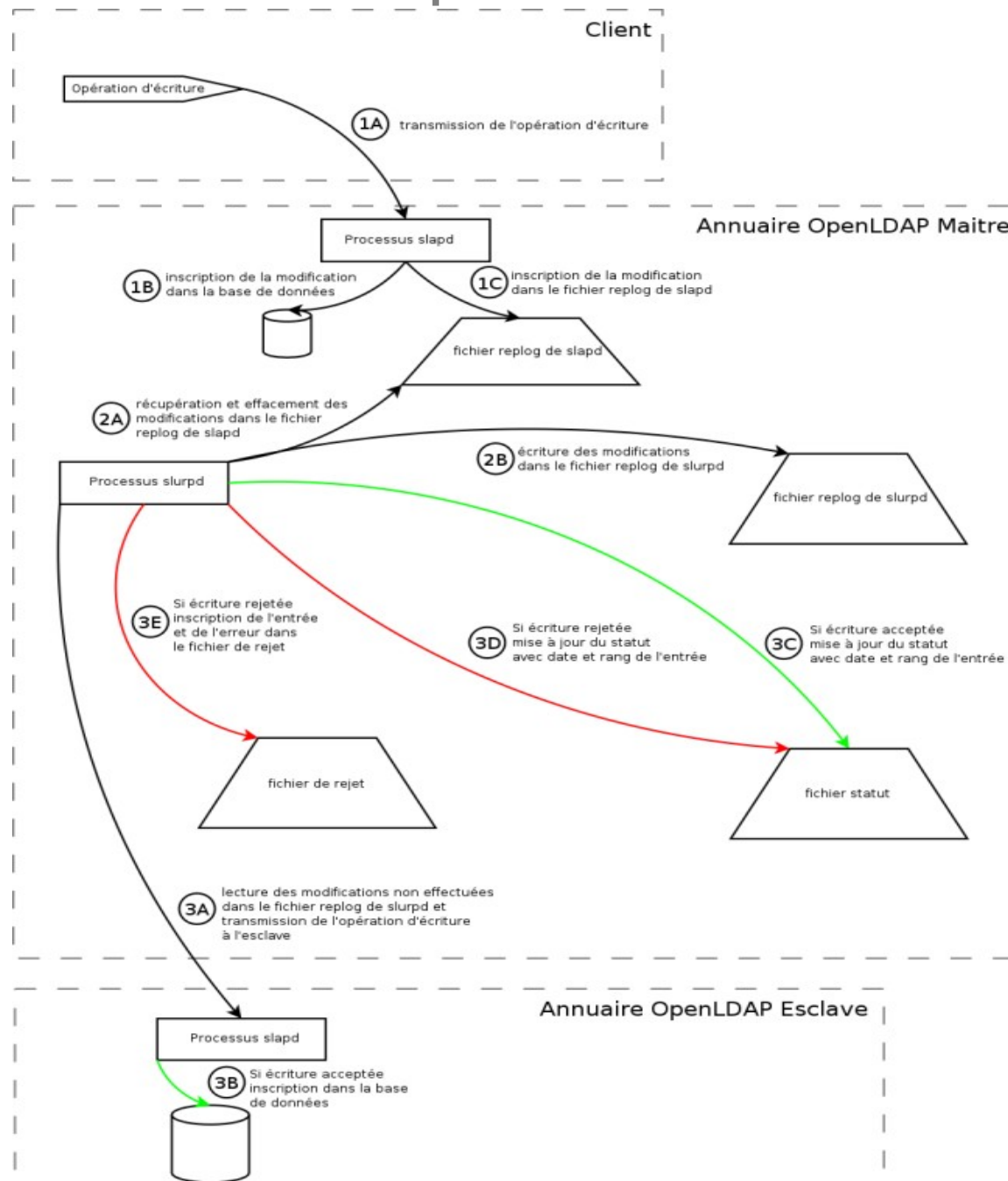
Plan de cours

- Présentation du logiciel OpenLDAP
- Installation
- Configuration
- Utilisation (client et serveur)
- Sécurité
- Schéma
- Réplication
- Performances

Types de réplication

- Réplication « push-based », ou SIR :
 - Server Initiated Replication : c'est l'annuaire maître qui réplique vers les esclaves
 - Disponible dans OpenLDAP avec slurpd ou une configuration particulière de Syncrepl
- Réplication « pull-based », ou CIR :
 - Consumer Initiated Replication : ce sont les esclaves qui établissent les requêtes vers l'annuaire maître
 - Disponible dans OpenLDAP avec Syncrepl (depuis la version 2.2)
- Réplication multimaîtres (multimaster) :
 - Plusieurs annuaires peuvent recevoir les écritures
 - Fonctionnalité expérimentale dans OpenLDAP

Fonctionnement de slurpd



Fichiers de slurpd

- Il y a 4 fichiers importants à analyser pour connaître le statut de la réplication slurpd :
 - Le fichier replog de slapd : il contient les modifications enregistrées sur l'annuaire maître mais pas encore récupérées par slurpd.
 - Le fichier replog de slurpd : il contient les modifications qui se trouvaient dans le fichier replog de slapd et qui ont été récupérées par slurpd.
 - Le fichier de statut : chaque ligne de ce fichier contient l'adresse et le port d'un annuaire esclave, ainsi que la date et le rang de la dernière modification acceptée ou rejetée.
 - Le fichier de rejet : il est spécifique à un esclave et contient les modifications rejetées.

Configuration de slurpd dans slapd.conf

- Un serveur est :
 - soit maître, et peut contenir une ou plusieurs directives définissant chaque esclave (destination de la réplication) :
 - replica
 - uri = URL de l'esclave
 - bind method = simple
 - binddn = DN de connexion à l'esclave
 - credentials = mot de passe
 - relogfile /var/lib/ldap/relog-hostname
 - soit esclave, et il contient deux paramètres lui permettant d'accepter les flux de réplication émis par le serveur maître :
 - updatedn : DN par lequel le maître propage les modifications
 - updateref : URL de renvoi lorsque des modifications lui sont adressées

Syncrepl

- Fonctionnalité apparue dans OpenLDAP 2.2
- Synchronisation basée sur des cookies : l'esclave va interroger le maître pour connaître les entrées différentes
- Deux modes :
 - refreshAndPersist : l'esclave (consumer) récupère en continu les modifications apportées au maître (provider) suivant l'état maintenu par le serveur
 - refreshOnly : l'esclave demande périodiquement les mises à jour au maître
- Syncrepl est compatible avec la configuration multimaîtres
- Possibilité de n'envoyer que les différences entre les entrées (c'est à dire au niveau des attributs) avec delta-syncrepl
- Possibilité de simuler la réplication SIR en utilisant un proxy LDAP (back-ldap)
- Réplication plus stable que slurpd car toujours convergente

Configuration de Syncrepl

- Sur le maître :

```
overlay syncprov  
syncprov-checkpoint 100 10  
syncprov-sessionlog 100
```

- Sur l'esclave :

```
syncrepl rid=123  
provider=ldap://provider.example.com:389  
type={refreshOnly|refreshAndPersist}  
interval=jj:hh:mm:ss  
retry=60 10 300 3 +  
searchbase="dc=linagora,dc=com"  
filter="(objectClass=organizationalPerson)"  
scope=sub  
attrs="cn,sn,ou,telephoneNumber,title,l"  
schemachecking={off|on}  
bindmethod=simple  
binddn="cn=syncuser,dc=linagora,dc=com"  
credentials=secret
```

Exercice

- Slurpd :
 - Configurer sur la même machine un annuaire maître et un annuaire esclave avec slurpd
 - Démarrer les processus
 - Valider la réplication
- Syncrepl :
 - Reproduire l'architecture précédente au moyen de SyncRepl
 - Valider la réplication

Plan de cours

- Présentation du logiciel OpenLDAP
- Installation
- Configuration
- Utilisation (client et serveur)
- Sécurité
- Schéma
- Réplication
- Performances

Premières mesures d'optimisation

- Compiler le noyau avec la configuration exacte de la machine
- Compiler le serveur LDAP avec le nouveau noyau activé
- Ne pas activer d'options de déboguage particulières ni d'historisation en production :
 - loglevel 0
- Dans le cas où l'historisation est activée, paramétrer syslog de façon à ne pas forcer la synchronisation sur disque de chaque message :
 - local4.* -/var/log/slapd.log
- Régler finement les options de paramétrage du backend utilisé

Sources classiques des problèmes

- Les pertes de performance en charge peuvent être liées :
 - au montant limité de mémoire qui arrive lorsque :
 - il y a trop d'index par rapport aux données ce qui fait basculer les opérations en partition d'échange disque (swap)
 - il n'y pas assez de processus ou de fils d'exécution (threads) par processus pour traiter les demandes
 - aux recherches « abusives » :
 - sur des attributs non indexés
 - générant trop de résultats
- Dans certains cas les limites systèmes doivent être augmentées pour éviter une indisponibilité en grande charge, notamment sur des instances utilisant un backend meta ou LDAP. Les limites correspondantes sont alors :
 - le nombre de fichiers ouverts maximum par processus
 - le nombre de fils d'exécution (threads) par processus

Pour optimiser, quelques principes ...

- Toute recherche doit être faite sur un attribut indexé
- En fonction du type de recherche, préciser les modes de recherches permet de limiter les index à mettre en œuvre (coût mémoire et CPU)
- Activer un cache DNS local (nscd) ou disposer d'un serveur DNS rapide
- Paramétrer le fichier DB_CONFIG
- Augmenter la quantité de mémoire vive disponible
- Pour les solutions plus complexes :
 - Mise en place d'un espace disque virtuel en mémoire
 - Dupliquer le nombre de serveurs et introduire de l'équilibrage de charge

Pour aller plus loin...

- Sites :
 - <http://www.openldap.org>
 - <http://www.commentcamarche.net/ldap/ldapinst.php3>
- Listes de diffusion :
 - ldap-fr@cru.fr (LDAP et OpenLDAP, en français)
 - openldap-software@openldap.org (en anglais)
- Livres :
 - LDAP : Administration système (Broché) de Gerald Carter
 - Annuaire LDAP (Broché) de Marcel Rizcallah

Et si vous souhaitez continuer votre apprentissage...

Détachez ce coupon et adressez-le au pôle **Formation** :

Yves MIEZAN EZO
 Email : formation@linagora.com
 Tél : 01 58 18 68 28
 Fax : 01 58 18 68 29



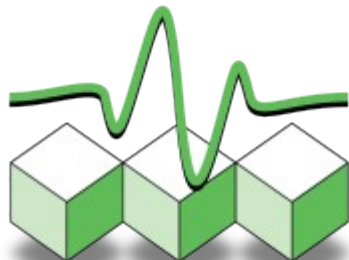
**Bénéficiez de
 100 €
 de réduction
 sur votre prochaine
 formation !**

Nom :
Prénom :
Société :
Mail :
Tél :
Stage :Date :
Tarif catalogue :€.....Réduction : - 100 €.....Tarif final :.....€

Code Opération « **LNGFetdevientfortenlibre** »

LINAGORA

Formation



Administration et sécurité

Merci de votre attention

LINAGORA *Formation*
formation@linagora.com