LDAP102: Creating the DSA

If your find these documents useful and feel the need to express that opinion in a tangible way, consider selecting an item from my Amazon Wish List.

awilliam@whitemice.org

Copyright

If you find this document useful or further it's distribution we would appreciate you letting us know.

Role Playing

- We are a climate control products distribution company called Ragnarok Service & Supply, which owns the DNS domain rss.nw.
  - We have a heterogeneous network
    - Windows 2000, XP, 9x, and Linux workstations.
      - No centralized authentication, each Linux system has its own user and password list, and Windows workstations are part of a workgroup and operate peer-to-peer.
        - There is a central fileserver named “Lif” on which all users have an account. The file server is a Samba box using clear text authentication.

Goals

- A centralized configuration
  - One user, one account, one password
    - Support for all clients
      - Windows domain for Windows workstations
        - The ability to use policies to control access to Windows workstations.
      - Authentication for LINUX workstations
      - Authentication for various services
        - IMAP, SMTP, Intranet, etc...
  - Accounts for extranet site
    - Customers coming to company website and signing on to an account and placing orders, etc...
  - Resolution of name space issues.
    - NetBIOS (WINS) vs. DNS.
  - Ability to make configuration changes without editing text files or using shell access to servers.

Details

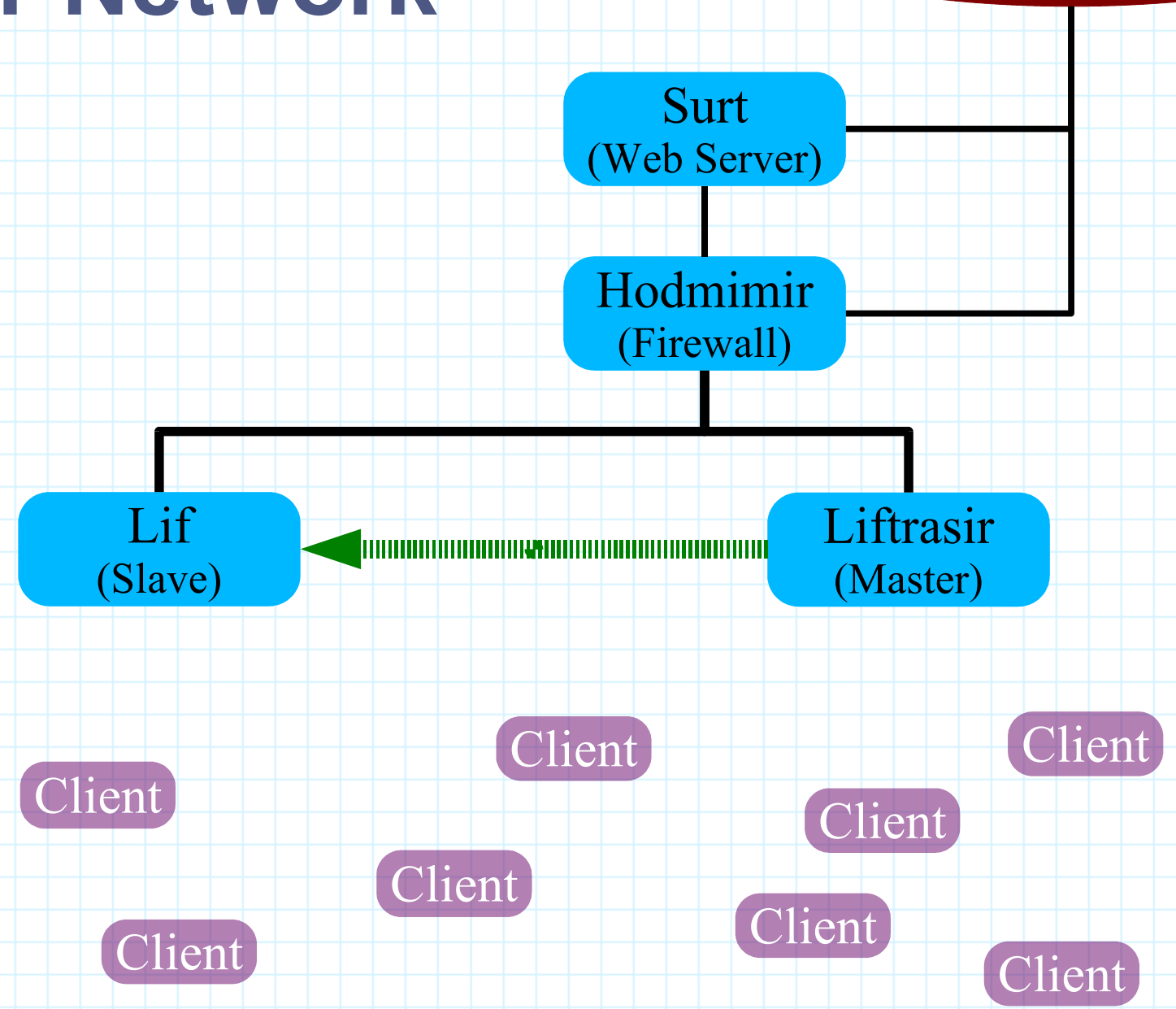
- Our new directory server will be named “Liftrasir”
  - A recent installation of SuSe Linux.
- Our current file server “Lif” will be configured as a slave replicant of “Liftrasir”, containing the full contents of the Dit.
- We will be using a recent release of OpenLDAP 2.2.x
  - We know Samba 4.x may implement its own internal DSA
    - We'll take this into account when designing our Dit

Assumptions

- You have a new machine setup and attached to your LAN with a static IP, this is “Liftrasir”
  - You have OpenLDAP 2.2.18 and appropriate dependent packages installed on “Lif” and “Liftrasir”.
  - No internal DNS or DHCP.
    - You're doing everything manually.

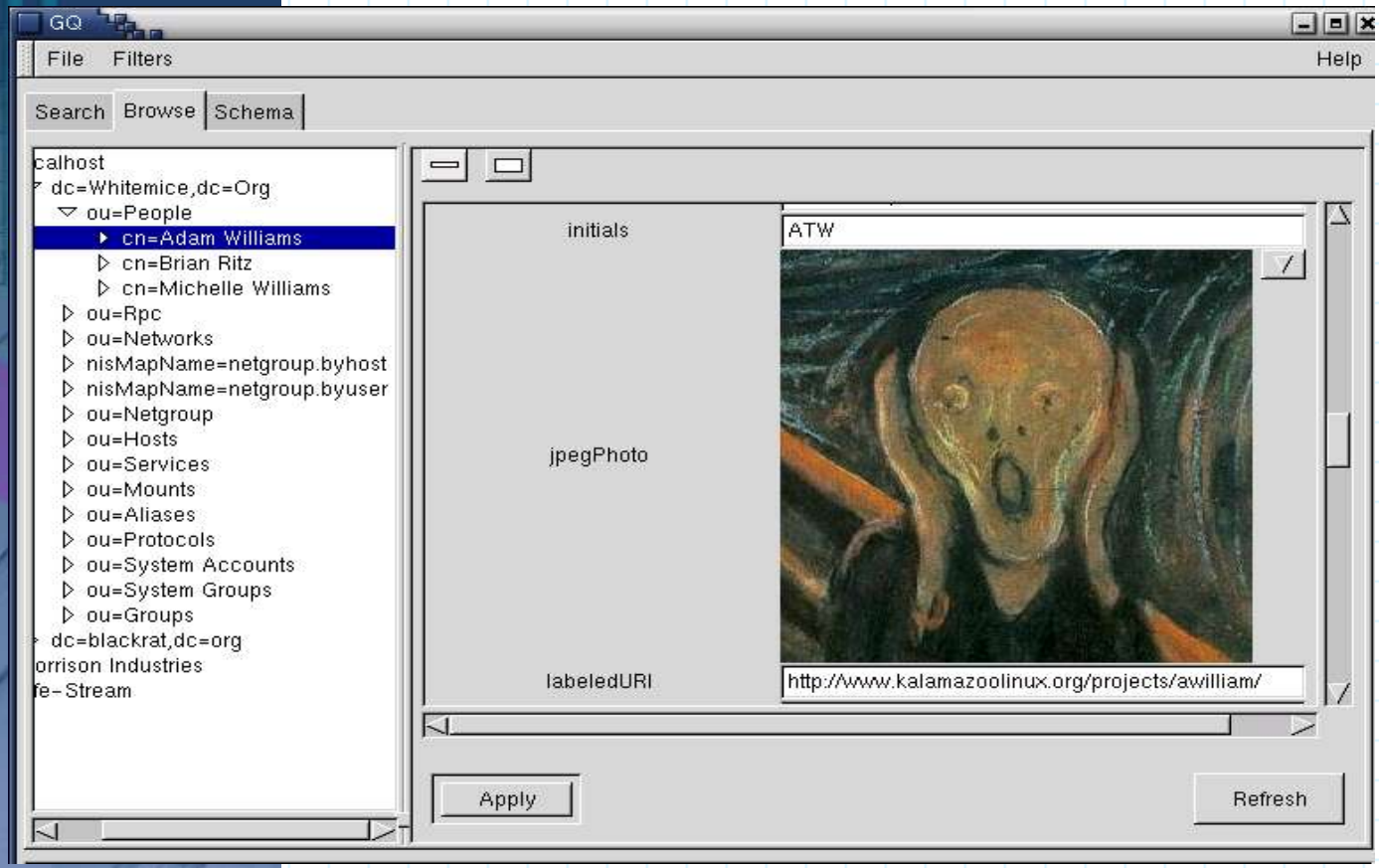
```
./configure --prefix=/opt/dsa --sysconfdir=/etc --localstatedir=/var/run/slapd \
--libexecdir=/opt/dsa/libexec --libdir=/opt/dsa/lib --mandir=/opt/dsa/man \
--sbindir=/opt/dsa/sbin --datadir=/opt/dsa/share --localstatedir=/opt/dsa/var \
--includedir=/opt/dsa/include --enable-aclgroups --enable-spasswd \
--enable-modules --enable-shared --enable-dynamic --with-tls \
--with-cyrus-sasl --enable-crypt --enable-ipv6=yes --enable-aci \
--enable-bdb --enable-rewrite --enable-ldap --enable-meta --enable-monitor \
--enable-ldbm --enable-sql --enable-lmpasswd --with-dyngroup \
--with-proxycache
```

Our Network


Big Scary World
Surt
(Web Server)
Hodmimir
(Firewall)
Lif
(Slave)
Liftrasir
(Master)
Client
Client
Client
Client
Client
Client
Client
Client

GQ

- We are also assuming you have a halfway decent general purpose LDAP client for editing the contents of the Dit.
  - There are no excellent general purpose LDAP clients.



- If you have a Linux workstation we recommend GQ although GQ is currently unmaintained and not without bugs.
  - Users of legacy platforms should try Jxplorer.
- We will not be covering how to use any LDAP client, thats up to you to muddle through.

Our Dit

- dc=rss,dc=nw
  - ou=Customers
    - cn=John Smith+o=Podunk Tire & Auto
  - ou=SubSystems
    - ou=BindSDB
      - zonename=rss.nw
      - zonename=168.192.in-addr.arpa
      - zonename=0.0.127.in-addr.arpa
      - zonename=_tcp.rss.nw
    - ou=ISCDHCP
      - ou=Configs
        - cn=rss-primary
      - ou=Servers
        - cn=Liftrasir.rss.nw
        - ou=Mail

RFC2247 Naming convention

Multi-Valued RDNs

Various IT Specific Stuff Tucked Away

Each service gets its own ou..

How these type of things look is very much dictated by the service.

A place to put objects (mostly groups) used to control access to the Dit itself.

Our Dit

- dc=rss,dc=nw
  - ou=Access Control
    - ou=Groups
      - ou=Replicants
      - ou=Administrators
      - ou=Marketing
    - ou=Entities
      - uid=lif
      - uid=cifsdc
  - ou=People
    - cn=John Quincy

A place to put objects (mostly groups) used to control access to the Dit itself.

We need to be able to grant special access rights to various groups.

Some entities only exist for the purpose of binding to the DSA, don't pollute the list of real accounts with these.

Many clients expect to find this, so we are going to fake it out via a "subordinate proxy"

Our Dit

- dc=rss,dc=nw
  - ou=SAM
    - ou=Groups
      - cn=backupops
      - cn=staff
      - cn=marketing
    - ou=Hosts
      - cn=lif
    - ou=Services
      - cn=smtp+ipServiceProtocol=tcp
    - ou=idMap
    - ou=Entities
      - ou=People
        - cn=John Quincy
      - ou=System
        - uid=root

Samba and NSS specific stuff we will put here.

This may become a special partition in Samba's special purpose LDAP server when Samba4 gets released.

Groups as in /etc/groups

Hosts as in /etc/hosts

Services as in /etc/services

The contents of /etc/passwd and then some.

The contents of /etc/passwd and then some.

Our Dit

- dc=rss,dc=nw
  - ou=Customers
  - ou=SubSystems
    - ou=BindSDB
      - zonename=
    - ou=ISCDHCP
      - ou=Configs
      - ou=Servers
    - ou=Mail
  - ou=Access Control
    - ou=Groups
    - ou=Entities
  - ou=People
  - ou=SAM
    - ou=Groups
    - ou=Hosts
    - ou=Services
    - ou=idMap
    - ou=Entities
      - ou=People
      - ou=System

- Every organizations Dit will look a bit different depending upon what applications and services they host or utilize as well as their organizational structure.

- Considerations
  - Applications and services hosted
  - Future direction
  - What clients expect to see
  - Controlling access

Configuring the DSA

```
include     /etc/openldap/schema.conf
include     /etc/openldap/access.conf
include     /etc/openldap/slapd.limits.conf
pidfile     /var/run/slapd.pid
allow       bind_anon_dn bind_v2 update_anon
include     /etc/openldap/database.conf
include     /etc/openldap/proxy.conf
database      monitor
```

Last entry creates the monitor database.

In a perfect world we wouldn't want to do this, but many Open Source applications don't get “it”.

Using include files allows us to break the configuration up over multiple files; and harder to trash in one fell swooop.

Configuring SASL

- We're going to start out just supporting PLAIN authentication.
  - We will migrate to something stronger later.

- OpenLDAP uses sasl-regexp expressions to locate an object that corresponds to a SASL authorization id.
  - A SASL authorization id in this context is a username.

```
sasl-secprops   none

sasl-regexp
  uid=(.*),cn=PLAIN,cn=auth
  ldap:///ou=SAM,dc=rss,dc=nw??sub?(&(uid=$1)(objectclass=account))

sasl-regexp
  uid=(.*),cn=rss,cn=PLAIN,cn=auth
  ldap:///ou=SAM,dc=rss,dc=nw??sub?(&(uid=$1)(objectclass=account))

security    ssf=0 transport=0 tls=0 sasl=0 update_ssf=0 update_
            transport=0 update_tls=0 update_sasl=0
```

Loading the schema

- Get the schema pack from KLUG's FTP server.
  - Unpack in /etc/openldap
    - Creates schema.conf which references schema files
    - Populates /etc/openldap/schema

```
include         /etc/openldap/schema/core.schema
include         /etc/openldap/schema/corba.schema
include         /etc/openldap/schema/cosine.schema
include         /etc/openldap/schema/inetorgperson.schema
include         /etc/openldap/schema/openldap.schema
include         /etc/openldap/schema/redhat/autofs.schema
include         /etc/openldap/schema/redhat/kerberosobject.schema
include         /etc/openldap/schema/dhcp.schema
include         /etc/openldap/schema/dnszone.schema
include         /etc/openldap/schema/evolutionperson.schema
...
```

ftp://ftp.kalamazoolinux.org/pub/projects/awilliam/ldap/schema.tar.gz

The Database

```
database        bdb
# Basic Information
suffix          "dc=rss,dc=nw"
directory       /var/lib/ldap
schemacheck     on
# Security Information
rootdn          "cn=Administrator,dc=rss,dc=nw"
rootpw          {SSHA}xxxxxxxxxxxxxxxx
password-hash   {CRYPT}
# Performance Directives
cachesize       10000
# Replication Directives
sessionlog      100 256
```

The "top" of the Dit stored in this database.

Where the database exists.

The "root" user of the database.

The "root" user's password.

How to encrypt user's passwords via the change password exop.

Number of entries cached in memory by the DSA.

Number of entries cached in memory by the DSA.

Indexes

- Indexes are configured in the database configuration file.

```
index   objectClass                                          eq,pres
index   uid,uidNumber,gidNumber,sambasid                     eq,pres
index   sambalogonTime,sambalogoffTime,sambakickoffTime      eq,pres
index   member,memberuid                                     eq,pres
index   sambapwdCanChange,shadowLastChange                   eq,pres
index   sambapwdMustChange,sambaprimarygroupsid              eq,pres
index   sambaacctFlags                                       pres
index   cn,mail,surname,givenname                            eq,sub,pres
index   roleoccupant                                         eq,pres
index   morrisondialaccess                                   eq,pres
index   morrisonvpnaccess                                    eq,pres
index   morrisonbranch,l                                     eq,pres
index   mailhost,mailroutingaddress,maillocaladdress         eq,pres
index   birthdate                                            eq,pres,sub
index   dhcpHWAddress,dhcpClassData                          eq,pres
index   zoneName,relativeDomainName,aRecord,pTRRecord        eq,pres
index   printer-name,rfc822mailmember,morrisonserialid       eq,pres
```

Limits

```
# Set query Size Limits & Search Timeouts
limits anonymous
  size.soft=512 size.hard=1024 size.unchecked=32767
  time.soft=10 time.hard=60
limits group="cn=Administrators,ou=Entities,ou=Access Control,dc=rss,dc=nw"
  size.soft=unlimited size.hard=unlimited size.unchecked=unlimited
  time.soft=60 time.hard=120
limits dn.exact="uid=syncrepl,ou=Entities,ou=Access Control,dc=rss,dc=nw"
  size.soft=unlimited size.hard=unlimited size.unchecked=unlimited
  time.soft=unlimited time.hard=unlimited
limits users
  size.soft=1024 size.hard=2048 size.unchecked=32767
  time.soft=15 time.hard=60
```

For whom

Limits

Access Control

rootDSE

```
access to dn.base="" by * read
access to dn.subtree="cn=monitor" by * read
access to dn.subtree="dc=rss,dc=nw"
  by dn.exact="cn=Administrator,dc=rss,dc=nw" write
  by group/groupOfUniqueNames/uniqueMember="cn=Administrators,ou=Groups,..." write
  by group/groupOfUniqueNames/uniqueMember="cn=Replicants,ou=Groups,..." write
  by * break
access to dn.subtree="ou=SAM,dc=rss,dc=nw"
  attrs=userpassword
  by dn.exact=”uid=cifsdc,ou=Entities,ou=Access Control...," write
  by self write
  by * auth
access to dn.subtree="ou=SAM,dc=rss,dc=nw"
  attrs=sambantpassword,sambalmpassword,sambapasswordhistory,sambabadpasswordcount,
sambabadpasswordtme,sambapwdcanchange,sambapwdmustchange
  by dn.exact=”uid=cifsdc,ou=Entities,ou=Access Control...," write
  by * none
....
access to * by * read
```

Access Control

- Defining access control (creating ACL declarations) is one of the most intimidating aspects of configuring the DSA.
  - The OpenLDAP Administration Guide
    - http://www.openldap.org/doc/admin22/slapdconfig.html#Access%20Control
  - Using Groups in ACLs
    - http://www.openldap.org/faq/data/cache/52.html
  - Regular expressions in ACLs
    - http://www.openldap.org/faq/data/cache/973.html
  - Sets in access control
    - http://www.openldap.org/faq/data/cache/1133.html
    - http://www.openldap.org/faq/data/cache/1134.html
  - The Experimental ACI Support
    - http://www.openldap.org/faq/data/cache/634.html
      - I don't recommend using ACIs, they just add yet-another-layer of access control, and thus another thing to consider when debugging access problems.

Preparing the Database

- OpenLDAP uses the Berkley database as the database.
  - A Berkley database consists of all the files in a directory.
    - Create you LDAP database directory (/var/lib/ldap) and make sure it is owned by the user the DSA operates as.
      - mkdir /var/lib/ldap
        chown ldap.ldap /var/lib/ldap
        chmod 770 /var/lib/ldap
    - Create the DB_CONFIG file inside the database directory.
      - The Berkley database reads this file for various performance directives.

```
set_cachesize      0 52428800 0
set_lg_regionmax   1048576
set_lg_max         10485760
set_lg_bsize       2097152
set_lg_dir         /var/lib/ldap/logs
```

Cache Size

Maximum Log File Size

Filename caching.

Cache size for transaction logs

Where to put the log files.

http://www.openldap.org/faq/data/cache/1075.html

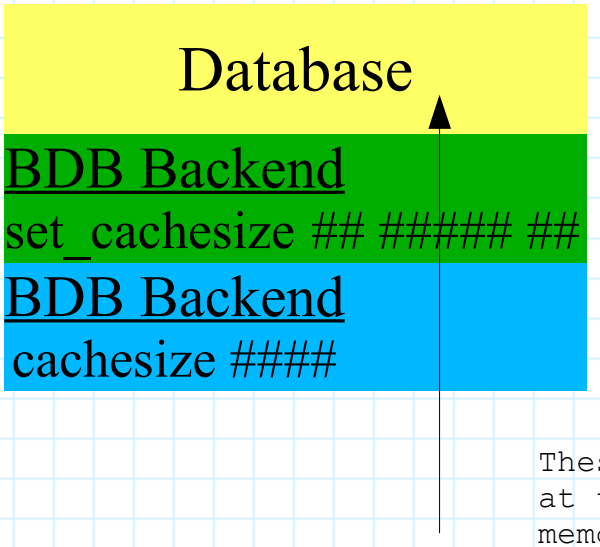
READ THIS!!!

DB_CONFIG

- Proper configuration and tuning of the Berkley backend is one of the most important aspects to producing a fast and stable DSA.
  - DB Backend Configuration
    - http://www.openldap.org/faq/data/cache/1073.html
  - I determine the proper BDB/HDB database cache size?
    - http://www.openldap.org/faq/data/cache/1075.html
  - The DB_CONFIG file
    - http://www.openldap.org/faq/data/cache/1072.html
  - Addition Berkley DB Configuration information at:
    - http://www.sleepycat.com/docs/
    - Selecting a cache size
      - http://www.sleepycat.com/docs/ref/am_conf/cachesize.html
- Be very cautious about third-party documentation
  - There is a great deal of bad or just obsolete advice floating about.
  - NEVER NEVER set "DB_TXN_NOSYNC"

NEVER

Caches



- One of the confusing aspects is that there are multiple data caches.
  - Summary: You use both.

These are two completely different caches and both are used at the same time. The BDB library cache controls how much memory, in bytes, is used by the library to maintain its internal bookkeeping information. The back-bdb entry cache controls how many LDAP entries to cache, independent of their size in bytes.
...
Fetching data from the BDB library is a lot slower than fetching from the entry cache. The BDB library locks data on a page level, and a page generally carries at least two data items in it, usually more. If multiple threads need access to data items that reside on the same page, they will be blocked and only one at a time will progress. The back-bdb entry cache stores one LDAP entry per cache entry. If multiple threads need access to multiple entries, they most likely will not interfere with each other.
You cannot use only one or the other, because the absence of either cache will slow down the entire system. You need both.
--Howard Chu

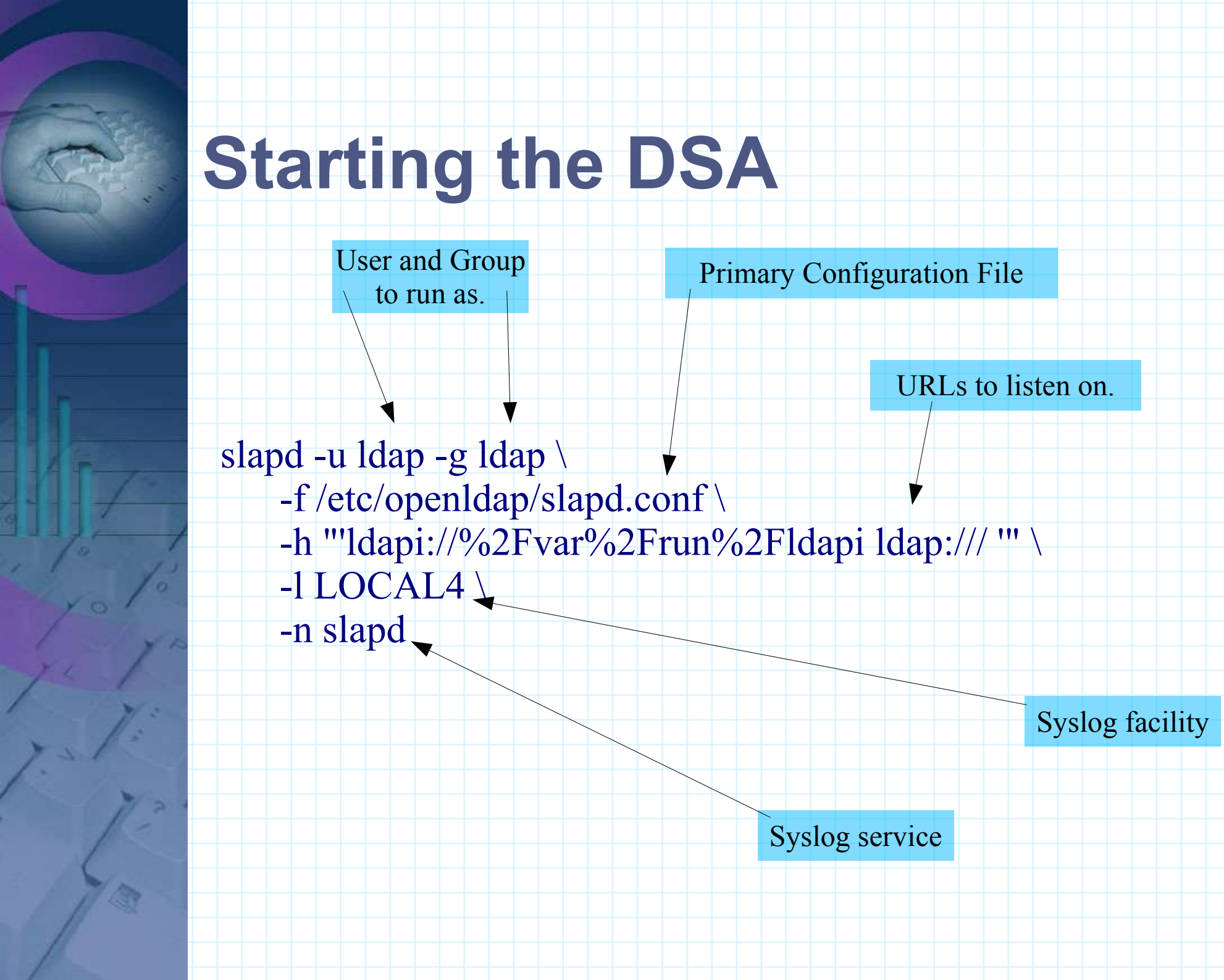
Creating the Dit

- Data can be loaded into the DSA via two methods.
  - Offline
    - The slapadd command loads LDIF data directory into the Berkley database.
  - Online
    - The ldapadd command loads LDIF data by connecting to a running DSA and performing the add/insert modifications.
- The bulk of the work in migrating to a directory-enabled network is migrating your existing data...
  - /etc/passwd, /etc/shadow
  - /etc/group
  - etc...
  - ... into the LDIF format.

Building the Dit

- Create your basic empty Dit with a text editor.
  - Load with slapadd
    - slapadd -n1 -p -v -l *tree.ldif*
      - You can increase the output level via -d *{level}*
    - Make sure the files are owned by the proper user
      - chown -R ldap.ldap /var/lib/ldap/*
  - Start the DSA
    - slapd -u ldap -f /etc/openldap/slapd.conf -h '"ldapi://%2Fvar%2Frun%2Fldapi ldap:/// '" -l LOCAL4 -n slapd
  - Convert existing data to LDIF
    - awk -f passwd.awk /etc/passwd > passwd.ldif
    - ...
  - Load the data into the Dit
    - ldapadd -D "cn=Administrator,dc=rss,dc=nw" \
      -x -W -v -f passwd.ldif
  - Data can be tweaked and cleanup via a general purpose LDAP client.

Starting the DSA



Converting the data

- A variety of scripts exist to convert flat files to LDAP
  - The most common is PADL's MigrationTools
    - http://www.padl.com/OSS/MigrationTools.html
  - Novell provides a general purpose CSV to LDIF conversion utility
    - http://www.novell.com/coolsolutions/netware/features/a_convert_to_ldif_nw.html
      - I haven't tried this.
- If your familiar with a scripting language it is often easier to write your own, since you know the peculiarities of your own data.
  - LDIF is a pretty straightforward format.
    - Start with DN
    - List objectclasses
    - List attributes + values
    - Terminate object with blank line

```
BEGIN {   FS=":"  }
{
 if ($3 >  99) {
  printf("dn: cn=%s,ou=People,ou=Entities,ou=SAM,dc=rss,dc=nw\n", $5);
  printf("objectclass: top\n");
  printf("objectclass: account\n");
  printf("objectclass: posixAccount\n");
  printf("objectclass: morrisonaccount\n");
  printf("objectclass: morrisonuser\n");
  printf("objectclass: person\n");
  printf("objectclass: organizationalPerson\n");
  printf("objectclass: inetOrgPerson\n");
  printf("objectclass: evolutionPerson\n");
  printf("objectclass: officePerson\n");
  printf("objectclass: mHybridPerson\n");
  printf("objectclass: morrisonperson\n");
  printf("cn: %s\n", $5);
  printf("sn: %s\n", substr($5, match($5, " ") + 1));
  printf("morrisonimallow: Y\n");
 } else {
     printf("dn: uid=%s,ou=System,ou=Entities,ou=SAM,dc=rss,dc=nw\n", $1);
     printf("objectclass: top\n");
     printf("objectclass: account\n");
     printf("objectclass: posixAccount\n");
    }
 printf("uid: %s\n", $1);
 printf("uidNumber: %s\n", $4);
 printf("gidNumber: %s\n", $3);
 printf("userpasswd: {CRYPT}%s\n", $2);
 printf("gecos: %s\n", $5);
 printf("homeDirectory: %s\n", $6);
 printf("loginShell: %s\n", $7);
 printf("\n");
}
END {}
```

- An awk script to convert the /etc/passwd file to LDIF.
  - Assuming “users” have a uidNumber greater than 99 and others are system accounts.

Converted Data

- /etc/passwd entry
  - awilliam:x:1000:100:Adam Williams:/home/awilliam:/bin/bash
    - becomes:

```
dn: cn=Adam Williams,ou=People,ou=Entities,ou=SAM,dc=rss,dc=nw
objectclass: top
objectclass: account
objectclass: posixAccount
objectclass: morrisonaccount
objectclass: morrisonuser
objectclass: person
objectclass: organizationalPerson
objectclass: inetOrgPerson
objectclass: evolutionPerson
objectclass: officePerson
objectclass: mHybridPerson
objectclass: morrisonperson
cn: Adam Williams
sn: Williams
morrisonimallow: Y
uid: awilliam
uidNumber: 100
gidNumber: 1000
userpasswd: {CRYPT}x
gecos: Adam Williams
homeDirectory: /home/awilliam
loginShell: /bin/bash
```

Setting up ldap.conf

- The OpenLDAP libraries read their defaults from the ldap.conf file, usually found in /etc/openldap
  - Not to be confused with the ldap.conf file used by PADL's NSS and PAM modules, usually found in /etc

```
URI     ldap://liftrasir  ldap://lif
BASE    dc=rss,dc=nw

#SIZELIMIT      12
#TIMELIMIT      15
#DEREF          never
```

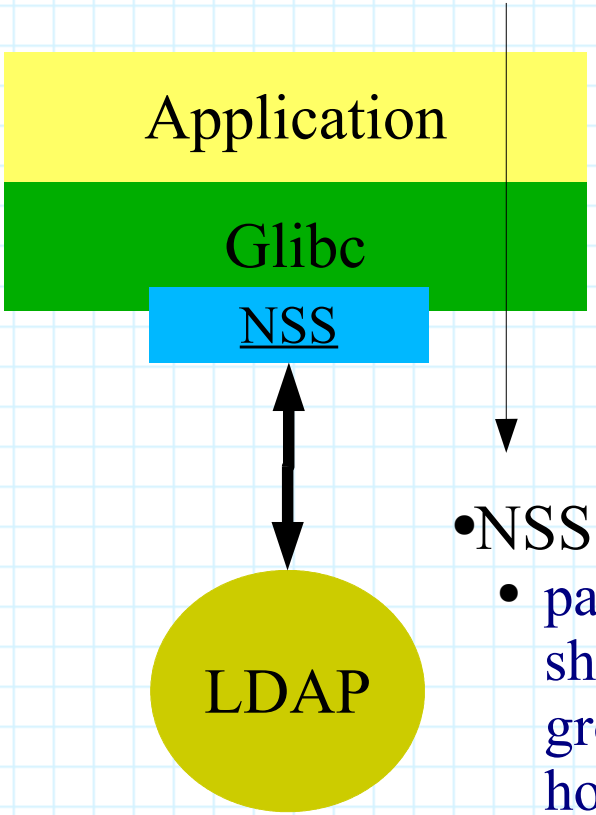
Default LDAP servers, in the order in which to attempt.

Defaults search base if none is specified

Testing with a query

- You should be able to query the DSA with the ldapsearch command.
  - Anonymous
    - ldapsearch -x -h localhost uid=fred dn
  - SASL (Plain text)
    - ldapsearch -W -h localhost -U fred -Y PLAIN uid=fred
      - -W option causes the user to be prompted for a password.

Configuring NSS



- Now that you have a working DSA you need to integrate it first with the underlying operating system.
  - Make the OS look at LDAP when a getpwent(...) call is made, rather than, or in addition to, looking in /etc/passwd.

- NSS operation is directed via /etc/nsswitch.conf.
  - passwd: files ldap
    shadow: files
    group: files ldap
    hosts: files ldap dns
    ethers: files
    netmasks: files
    networks: files ldap
    protocols: files ldap
    rpc: files ldap
    services: ldap files

Setting up NSS

- PADL's NSS module is configured via the ldap.conf file usually found in the /etc directory.

- This file is extremely well commented.

/etc/ldap.conf:

```
host liftrasir.rss.nw lif.rss.nw
base dc=rss,dc=nw
ldap_version 3
#binddn cn=proxyuser,dc=example,dc=com
#bindpw secret
rootbinddn uid=nss,ou=Access Control,dc=rss,dc=nw
pam_password exop
nss_base_passwd    ou=Entities,ou=SAM,dc=rss,dc=nw?sub
nss_base_group     ou=Groups,ou=SAM,dc=rss,dc=nw?one
nss_base_hosts     ou=Hosts,dc=rss,dc=nw?one
.....
```

Testing NSS

- Once NSS has been configured, remove a user from the /etc/passwd file and issue the id command to lookup information about the user.
  - id fred
    uid=1000(fred) gid=100(users) groups=100(users),16(dialout),33(video)
    - If the user name is valid, and you can query the DSA, recheck your NSS configuration.

Setting up PAM

- Most current distributions offer an administrative tool to select authentication via LDAP.
  - Some distributions use the pam_unix2 module, recent versions of which perform LDAP authentication automatically if NSS LDAP is configured.
  - Otherwise you need to add pam_ldap to the appropriate points in your PAM stacks.
    - ftp://ftp.kalamazoolinux.org/pub/pdf/pam_and_nss.pdf
  - Once PAM is configured you should be able to login 'normally' just as if you had an account in /etc/passwd.

Create Slave

- Configure the DSA on our replicant (lif) the same as for liftrasir.
  - Except
    - We don't load any data
    - We remove the sessionlog entry
    - We create a synrepl stanza.

database.conf

```
sessionlog          100 256
syncrepl rid=52
  provider=ldap://liftrasir.rss.nw:389
  type=refreshAndPersist
  searchbase="dc=rss,dc=nw"
  filter="(objectclass=*)"
  scope=sub
  schemachecking=off
  updatedn="uid=synrepl,ou=Entities,ou=Access Control,dc=rss,dc=nw"
  bindmethod=simple
  binddn="uid=lif,ou=Entities,ou=Access Control,dc=rss,dc=nw"
  credentials=*********
```

- When the slave DSA is started it will automatically load itself with data from the master DSA.

Backup

- You can take a safe atomic snapshot of your Dit with the slapcat command.
  - Produces an LDIF to standard output or a file.
  - slapcat -n1 -l snapshot.ldif