Outline

- Background
- Motivation
- Boost
- Case Study
- Recent Developments
  - Overall cleanup & redesign of proxies & libldap
  - Reliability of connections
  - Identity management & propagation
- Future Trends
Background

- **Back-ldap**: OpenLDAP 1.2  
  - Independently developed @ Symas & SysNet  
  - @ SysNet: proxy public DSA from behind a firewall  
  - BTW, it was our first build on AIX

- **Back-meta**: OpenLDAP 2.0 (made it into 2.1)  
  - Proof-of-Concept to glue heterogeneous DSAs  
    (1 OpenLDAP 1.2 for ~10,000 users, many ADs and Lotus Notes for ~50,000 users) during largest national bank merge (3 major banks merging into IntesaBCI, then Bancaintesa)  
  - Allowed SysNet to become supplier of that group and implement merging of LDAP, SMTP, IMAP, Web Services & more...
Background

- OpenLDAP (2.0, + Sendmail, Cyrus IMAP, ...): in 2001 allowed SysNet to seamlessly deliver/rewrite ~60,000 mailboxes with 3 major domains, dozens of minor domains, distributed all over the world, innumerable, layered aliases from previous mergers, remapping all outgoing message addresses with the new address (@intesabci.it).

- In 2002, after the merger was done, all email addresses (and mailboxes) had to switch to a new common domain (@bancaintesa.it), maintaining support for all old addresses.

- It all happened one night (at about 2:00am; but I, with few colleagues, didn't sleep for the rest of the night, waiting for trouble to come): 0 sec. downtime (for about one hour, SysNet's SMTP server was the primary MX for the whole stuff, and it took the whole day to dispose of the queue...).
Motivation

- DSAs are often distributed in nature
- IT wild growth favored parallel intra-department development of independent DSAs
- Contrasting needs to **connect/centralize** services, global administration & security, and **distribute** responsibilities, roles, local administration for **reliability** and **performance** (e.g. Italian Local Health Care Administration “AUSL Parma”: 2500 users on 4 sites)
- **Alternative solution to growth in DB size.**
OpenLDAP 2.2 introduced **overlays** *(thanks to Howard Chu and Symas!)*

Another key evolution was the **common interface** for backend operation call

```c
typedef int (*BI_op_func)(Operation *op, SlapReply *rs);
```

Lots of development moved into designing overlays, either **general purpose** or **custom**
Boost (cont'd)

• There's a joke at SysNet:

  any time someone asks how to solve a problem, no matter what, the answer invariably is:

  “Ando can prepare an overlay”...

  (or Luca, or Marco, or...)
(Anonymous) Case Study

- We recently started a big (I mean: **BIG**) development work on LDAP proxying; it's about to enter production
- Key idea: the proxy must exploit any knowledge of data location (the “right” target) to:
  - *maximize selected performances*, and
  - *minimize traffic*
- **Back-ldap** is ideal when the target is **known**
- **Back-meta** is ideal when **no knowledge** can be gathered
- OpenLDAP (2.3) already contained **all** (well, actually most of) the required **ingredients**
(Anonymous) Case Study

- Solution (guess what): an **overlay**...
  - Analyze requests
  - Guess the “right” target
    - Guessed? Use **back-ldap**
    - Not guessed? Fallback to **back-meta**
    - But not just regular back-meta: for example, do not re-contact the targets already explored during the guess
    - Use different error handling for direct/broadcast searches
    - ...

October 13, 2006
(Anonymous) Case Study

- Custom overlay: intercepts and analyzes requests
- Built on top of stock `back-meta`
- Contains a private instance of `back-ldap` for each target, plus
- a private instance of `back-meta` for non-privileged targets
- Caches information about specific data location
- Provides configurable “smartness” to infer data location from request data (via `librewrite`)
- Specially handles bind & write
- Implements cross-target rename using “relax” control
- Introduces and uses `back-monitor` customization
- ~15,000 lines of code (plus changes to mainstream code)
(Anonymous) Case Study
Recent Devel: Cleanup & Rework

- OpenLDAP proxy and libldap code benefited from that project
- When the project started, `test036-meta-concurrency` could hardly start, and would core after few operations
- Concurrency issues in back-ldap, back-meta, but significantly in `libldap`
- Work occurred in:
  - Abandon
  - Connection
  - Request/response
  - Referrals (still issues there; proxies directly handle referrals)
Recent Devel: Reliable Connections

Added features:

- “conn-ttl” to expire cached connections after some time
- “idle-timeout” to expire inactive cached connections
- (Customizable) URI list rearranging when connection fails;
  - this is now *monitorable* (in back-ldap)
  - and *manageable* to alter run-time the list of URIs
- “quarantine” to avoid trying to recontact a target URI that does not respond, with configurable retry pattern
- Customizable URL selection when chasing referrals (previously tried first to last)
Recent Devel: Reliable Connections

uri "ldap://host1 ldap://host2 ldap://host3" → contact host1

success? yes

URI list remains the same

URI list reflects connection attempts result

uri "ldap://host2 ldap://host3 ldap://host1" → contact host2

success? no

yes
**Recent Devel: Search Broadcast**

Parallelize connection establishment and searches
- Appreciable first-search improvements with slow targets

<table>
<thead>
<tr>
<th><strong>old</strong></th>
<th><strong>new</strong></th>
</tr>
</thead>
</table>
| for each target {
  if (!bound) ldap_bind();
} for each target {
  ldap_search();
} for each target {
  switch (ldap_result()) {
   case searchResult*:
    handle(); break;
  }
  for each target {
    handle errors;
  }
} | for each target {
  if (bound) ldap_search();
  else ldap_bind();
} for each target {
  ldap_search();
} for each target {
  switch (ldap_result()) {
   case bindResponse:
    ldap_search(); break;
   case searchResult*:
    handle(); break;
  }
  for each target {
    handle errors;
  } if (binding) ldap_unbind();
} |

October 13, 2006
Recent Devel: Identity Management

Added features:

- “idassert” extended to back-meta (supersedes “pseudoroot”)
- Defer binds until required
- Use “idassert” to restore broken connections seamlessly
- Back-ldap “acl-bind” and “idassert-bind” clarification:
  - “acl-bind”: identity for privileged operations
  - “idassert-bind”: identity for proxy authorization
- “idassert” supports obsolete proxy authz implemented by others and pre-RFC 4370 encoding interpretation like in mozilla SDK: “obsolete-proxy-authz” & “obsolete-encoding-workaround”
Recent Devel: Identity Management

Added features (cont'd):

- "idassert" is used by slapo-chain for **auth'd** referral chasing (for example, test007, test017 and more use it to allow writing to a slave and have modifications proxied to the master)
- Separate "idassert" can be configured for **well-known** URIs
- Unknown URIs can only be proxied anonymously
- Connections for referral chasing can be **cached** much like in regular back-ldap (slapo-chain **is** regular back-ldap, all in all...)
Future Trends

- Unlike I announced some time ago, **back-meta is not dead**; development will continue.
- Back-monitor has been extended by providing an API to register custom entries or adding data and callbacks to existing entries; back-bdb/hdb and back-ldap already use this feature.
- Monitor cached connections: local DN, bound DN, status, ... 
- Add back-config support to back-meta.
- Slapo-rwm suffers from design limitations; it needs extensive work to overcome them (and back-config support...)
- Back-ldap will support “distributed procedures” implementation (designed as an overlay resembling slapo-chain) (**ralf**)
- BTW, back-ldap can now be used to **push** syncreplication (**hyc**)
Conclusions

- Proxying is strange: most of the times data just passes thru.
- What makes it “easy” is that it has little to do with reliability:
  - A proxy could be disk-less, while a database couldn't.
  - If a disk crashes it's a pain; if a connection breaks, just retry.
- Often we need proxies because of poorly designed db/hw.

- The fact that SysNet's business (at least: customers' demands) about LDAP proxying grows instead of decreasing could mean that poorly designed db/hw are growing as well? :)
- In any case OpenLDAP proxy solutions seem to be very flexible.