Leveraging RFC 4533 to build a heterogeneous LDAP server replication system

Introduction

Replication has always been one of the most critical part of a production ready LDAP server. It guarantees the fail-over, and the scalability to a certain extent. Many replication mechanisms are possible, and OpenLDAP has implemented more than one since its first version.

We will analyze the RFC 4533 proposal (Syncrepl), and see how we can leverage this RFC in order to build a full replication system that works across servers.

We will also try to show how such a system can be used for more than just doing replication.

A bit of history

X.500 is the base, and LDAP directly inherits from it. Replication was a part of X.500 specification, as it describes a distributed directory service.

Back then, X.525 described two modes of replication :

- caching
- shadowing

When LDAP was first defined, replication was not part of the specifications, as LDAP was only supposed to be a Protocol. It turned out that implementing a LDAP server was easier than writing a X.500 server, and the first LDAP servers quickly appeared.

Many proposals has been published about replication or related to replication. Some are RFCs, some are drafts :

- LDAPv3 Replication Requirements (RFC 3384) October 2002
- LDAP entryUUID Operational Attribute (RFC 4530) June 2006
- LDAPv3 Content Synchronization Operation (RFC 4533) June 2006
- LDAP Multi-Master Replication Protocol (draft mult-mast-repl, November 1997)
- Mandatory LDAP Replica Management (draft LDUP-mrm, March 2003)
- General Usage Profile for LDAPv3 Replication (draft ldup-usage-profile, September 2003)
- LDAP Replication Architecture (draft LDUP-model, October 2003)
- LDUP Update Reconciliation Procedures (draft LDUP-urp, october 2003)
- The LDUP Replication Update Protocol(draft LDUP, December 2003)

And in a controversial paper, **Kurt Zeilenga** almost killed LDUP :

- LDAP Multi-master Replication Considered Harmful (draft LDUP-harmful, february 2004)

(this draft has been discussed by many implementors who basically say Kurt is theoretically right, but practically, it has no impact on real life [1], [2])

As of today, replication is still not part of the LDAP specifications. OpenLDAP, for instance, has implemented slurpd, which is a Master-Slave replication system, when other LDAP server are using other approaches, allowing a Multi-Master replication. But OpenLDAP has also implemented another replication system, based on a new specification, RFC 4533, or 'LDAP Content Synchronization Operation'.

Last, not least, we should also consider that replication relies on a communication protocol, and this protocol is not stabilized yet : syncrepl is a bit verbose, and delta-syncrepl is still a draft.

RFC 4533, what's in the box ?

RFC 4533, or “The LDAP Content Synchronization Operation” describes a protocol by which two LDAP servers can replicate by using an extended version of a **search** operation. The protocol itself is called **syncrepl**.

This RFC was issued back in June 2006, which is quite a short time ago, and was implemented by **OpenLDAP**.

So what does it bring that other replication system does not offer ?

- It's using the LDAP protocol to execute replication
- The servers don't have to maintain an history of changes
- It can detect when a full content reload is necessary
- It can be used in both Master-Slave or Master-Master replication systems
- It offers two modes : Polling or Listening

It is also supposed to fix some of the existing replication problems (taken from the RFC):

- Failure to ensure a reasonable level of convergence
- Failure to detect that convergence cannot be achieved (without reload);
- Require pre-arranged synchronization agreements
- Require the server to maintain histories of past changes to DIT content and/or meta information
- Require the server to maintain synchronization state on a per-client basis
- Overly chatty protocols.

RFC 4533 is supposed to replace the defunct LDUP group drafts, and it is, in fact, the first RFC specifically describing a common replication system.

How can we use this RFC in an heterogeneous LDAP environment ?

The beauty of this RFC is that it relies on the **LDAP** protocol, allowing an implementor to build his own system, and still being able to replicate with another server (at least, theoretically).

Apache Directory Server team has worked for years on a Multi-Master replication system called **Mitosis**, based on a proprietary protocol, and using a database to store an history of changes. At some point, it worked, but was not reliable, not performant, and not interoperable.We also assume that it was never deeply tested, and had a lot of problems.

We decided to switch to RFC 4533 as OpenLDAP implemented this RFC successfully, and demonstrated that it was safe and efficient.

The idea was to implement this system into Apache Directory Server as the base replication mechanism, replacing **Mitosis**, but also to allow a replication between **OpenLDAP** and apache Directory Server, as we may want to have both system present in an IT system.

What for ?

Apache Directory Server and OpenLDAP are two open source LDAP servers, and our both teams are collaborating pretty well. However, both server have different characteristics. Typically, Apache Directory Server is a Java LDAP server, which has been designed to be embeddable into Java applications. It makes sense that those applications use a subset of a global set of data handled by a centralized LDAP server (one can imagine a remote application which is not always connected to the company network).

Another aspect is that Apache Directory Server implements Triggers and Stored procedures [3], allowing this server to be a perfect e-provisioning system. We can have a central OpenLDAP server holding all the entries, and Apache Directory Server being a slave reacting to changes by spreading those changes to all the systems it is connected to (sending mails, calling web-services, ...)

We can also imagine two companies using different LDAP servers that does not want to merge their data in one monolithic LDAP server (simply because they are not working in the same area, they are just collaborating). Being able to share the data through a common replication mechanism can be seen as a big benefit for both those companies.

Last, not least, we can imagine to use this protocol to build a fast an secure system to do auditing, or backups, across many different LDAP servers, instead of dumping huge LDIF files and processing them.

Even if those scenarii seem a bit hypothetical as of today, it may perfectly become a reality in the next few years.

Apache Directory Server Roadmap

OpenLDAP has already a working RFC 4533 replication system implementation, it's time for Apache Directory Server to catch up !

Last months were pretty busy for us, as we had to build all the elements we needed to be able to implement this RFC :

- remove Mitosis from ourcode base
- include support for entryUUID and entryCSN
- implement a journal to be able to efficiently implement synrecpl
- define a client-API being able to communicate using LDAP protocol with a remote server
- implement the needed controls (SyncRequest, SyncInfo, SyncDone, SyncState)
- implement the consumer part
- write a prof of concept, with ADS being a consumer and OpenLDAP a producer

What remains to be done is :

- write the producer part
- write the conflict resolution
- write the integration tests
- install a platform to do non-regression tests with OpenLDAP and ADS

Future steps

OpenLDAP has started to work on a new protocol, less verbose, called **DeltaSyncrepl**. This protocol only transfer the modified data, not all the entries. This is most certainly the way to go, and we have to move to this protocol as soon as we have a working **Syncrepl** system.

Another step would be to see other servers implementing **Syncrepl** as a standard available system, at least the producer part of the protocol.

We should also consider replication of the schema, and how to allow cross replication of systems where schemas might potentially differ. This could alleviate the need of virtual directories, up to a point

Conclusion

There is still a lot to do on our side, but we are pretty confident that **syncrepl** is the road to follow. We expect that this replication system will be adopted by other LDAP server as well, to a point it becomes a must have.

From Apache Directory Server point of view, this is the best opportunity to demonstrate that the features we have implemented into the server can be used in a mixed environment, not forcing our users to bet everything on a single horse !

References

[1] http://directory.fedoraproject.org/wiki/MMRConsideredHarmful
[2] http://blogs.sun.com/DirectoryManager/entry/read_only_replicas_considered_harmful
[3] http://directory.apache.org/community%26resources/ldap-stored-procedures-and-triggers-in-apacheds.html