



Deploying MySQL Cluster as a High-Availability Data Store for OpenLDAP

An OpenLDAP Technical Whitepaper by Symas

Table of Contents

| | |
|--|----------|
| 1 INTRODUCTION..... | 3 |
| 2 BRINGING UP MYSQL CLUSTER CARRIER-GRADE EDITION..... | 3 |
| 3 PREPARING OPENLDAP TO USE MYSQL CLUSTER CARRIER-GRADE EDITION | 4 |
| 3.1 Configuring a MySQL Cluster Carrier Grade Edition Database in OpenLDAP..... | 5 |
| 4 CONVERTING AN OPENLDAP DATABASE FROM HDB/BDB TO THE DRIVER FOR MYSQL CLUSTER CARRIER-GRADE EDITION..... | 6 |
| 5 FURTHER CONSIDERATIONS..... | 6 |
| 6 REFERENCES..... | 7 |
| 7 ABOUT SYMAS | 7 |
| 8 ACKNOWLEDGEMENTS..... | 8 |

1 Introduction

The integration of an OpenLDAP driver to a MySQL Cluster Carrier-Grade Edition database back-end is a substantial step forward in directory database technology for the largest and most dynamic directory deployments:

- First and foremost, directories can now be integrated which would have required configurations costing much more than the original application business case could support.
- Second, directories can be much more flexibly shared and their storage configured independently of the running OpenLDAP servers.
- Finally, directory data can be stored in a form that is relatively easy to access from MySQL as well as through LDAP itself.

There are cost and performance trade-offs to be made when you deploy MySQL Cluster as the directory database for OpenLDAP. There is a big improvement in cost and flexibility which is counterbalanced by modest costs in performance and complexity. It is highly likely that larger directories (either in storage size or in complexity of client applications or network configuration) with high update rates will best benefit from this new database option.

Because the OpenLDAP Driver for MySQL Cluster Database Backend is a fully transparent database implementation, it is quite easy to migrate an OpenLDAP Hierarchic Database (HDB) database to MySQL Cluster. OpenLDAP's HDB is an evolutionary extension of its original Berkeley Database (BDB) back-end which is still widely in use, and which provides excellent data storage capability for many directory users and applications. Generally, the discussion in this document should apply to directories using either of those back-ends.

This white paper discusses the practical project elements of a migration from OpenLDAP's HDB database backing storage technology to MySQL Cluster. This is the natural starting point for scenario planning. It is a relatively trivial matter to bring up OpenLDAP using its older database technology and implement a fully functional test-bed or even smaller and simpler production system. Introduction of the Driver for MySQL Cluster Edition is then transparent to all but the OpenLDAP and MySQL database administrators.

2 Bringing up MySQL Cluster Carrier-Grade Edition

There is an extensive range of information available for bringing up MySQL Cluster Carrier-Grade Edition. The core documentation is available on the Web:

* Product Documentation: <http://dev.mysql.com/doc/refman/5.1/en/mysql-cluster.html>

* Getting Started Guide: <http://www.mysql.com/products/database/cluster/get-started.html>

* Configuration Guide: <http://dev.mysql.com/doc/refman/5.1/en/mysql-cluster-configuration.html>

There are no unique or special requirements for hosting an OpenLDAP database on MySQL Cluster. Normal configuration planning and tuning approaches should provide excellent performance.

While preparing, you will find it useful to use the online sandbox and/or configurator:

<http://www.severalnines.com/sandbox>
<http://www.severalnines.com/config>

There is extensive documentation at the Web locations above including a quick-start guide that contains a lot of useful information on getting started with MySQL Cluster:

<http://www.severalnines.com/blog/openldap.php>

Note also that during set up, you will need to make sure that the `MY.CNF` and `CONFIG.INI` files correctly identify the data directories (`DATADIR`). Also, note that you need to coordinate the number of database connections (data nodes) OpenLDAP expects to be using with the number defined in `CONFIG.INI`. The value specified in the `olcldbconnections` (`dbconnections` in `slapd.conf` style configuration) must be no greater than the number of MySQL daemons (`mysqld`) defined in `CONFIG.INI`.

3 Preparing OpenLDAP to Use MySQL Cluster Carrier-Grade Edition

Presumably, since you are planning to migrate an existing OpenLDAP directory database from a prior installation you have experience in installing and maintaining the OpenLDAP Software. There are two options open for you to upgrade your existing OpenLDAP software to prepare to run MySQL Cluster Carrier Grade Edition:

- build an updated version from the source code on the OpenLDAP Project Web site (<http://www.openldap.org/>)
- obtain a copy of the Symas OpenLDAP package with the Driver for MySQL Cluster Carrier Grade Edition (`back-ndb`) prebuilt for you.

Many enterprise users of OpenLDAP are comfortable building OpenLDAP software from the source code. This involves downloading the source code, applying patches and updates to the OpenLDAP and other associated Open Source software projects as suggested by the project, and building the software using the tools provided/available. This is best done by someone with significant experience with the platform for which your enterprise needs OpenLDAP and with Open Source software. However, this approach gives the enterprise a great deal of flexibility and a valuable internal capability to take advantage of the evolution of OpenLDAP software.

If you choose to build the OpenLDAP software yourself, the OpenLDAP Project strongly recommends working with the official Project Release sources. There are many updates available in the source code library that have not been included in official Released source and unless you are especially deeply involved with the Project and competent to do the necessary testing we would advise you to stick with the Release source tarballs. OpenLDAP Releases go through extensive testing prior to Release. You should take advantage of the assurance that provides that the software is ready for production use.

The primary documentation for OpenLDAP is the OpenLDAP Admin Guide which can be downloaded from <http://www.openldap.org/doc/admin> on the Web. Chapter 4 of the Admin Guide is "Building and Installing OpenLDAP Software". In addition, you should refer to the documentation, mail lists, and forums for the related Open Source Software projects (particularly OpenSSL and Cyrus SASL).

During the actual software build process, you will need to direct the process to the appropriate MySQL Cluster 7 libraries. You will need to identify the MySQL client libraries and the NDB API library.

These must be from the same release of MySQL Cluster Carrier Grade Edition 7 but may be different than the releases running on data nodes in the network (though it is always best to synchronize).

There is another option to building OpenLDAP Software yourself. Symas offers commercial licensees of MySQL Cluster Carrier Grade Edition access to a version of Symas OpenLDAP including binary modules pre-built for the back-ndb database. Symas OpenLDAP is a convenient binary distribution of OpenLDAP and the related software, including in this case, the driver (`back-ndb`) module. Symas OpenLDAP is available for many different processor architectures and software platforms

3.1 Configuring a MySQL Cluster Carrier Grade Edition Database in OpenLDAP

There are a number of parameters that tell OpenLDAP how to access the MySQL Cluster Carrier Grade Edition data store(s). These parameters are specified either in the OpenLDAP `slapd.conf` configuration file (an older deprecated capability) or through LDAP Data Interchange Format (LDIF) commands which update the internal `cn=config` database. In the `slapd.conf` file, these parameters would be coded one per line after the database directive for the database. An example of that is shown in Listing 1.

Listing 1 – Example of `slapd.conf` configuration for OpenLDAP Driver for Mysql Cluster (`back-ndb`)

```
# What follows is a sample definition of the configuration of a back-ndb
database
database          ndb
suffix            "o=example"
suffix            "dc=example,dc=com"
rootdn            "o=example"
rootdn            "dc=example,dc=com"
rootpw            secret
dbhost            localhost
dbport            3306                # 3306 is the default
dbconnect         localhost
dbuser            root
dbpass            secret
dbname            OpenLDAP
# dbconnections comment: (you need one free [mysqld] in
#                   config.ini for each dbconnection
dbconnections     1                # 1 to n
dbsocket          /tmp/mysql.sock   # for localhost use
```

Listing 2 shows the same information in LDIF format. This would be added to the configuration using the `ldapadd` command after the basic OpenLDAP server (`slapd`) was operational.

Listing 2 – Example of `cn=config` configuration for OpenLDAP Driver for Mysql Cluster (`back-ndb`)

```
# What follows is a sample definition of the configuration of a back-ndb
database
# First we define the back-end
dn:                olcBackend=ndb,cn=config
objectClass:       olcBackendConfig
olcBackend:        ndb
dn:                olcDatabase=ndb,cn=config
objectClass:       olcDatabaseConfig
objectClass:       olcHdbConfig
olcDatabase:       ndb
```

```
olcSuffix:      dc=example,dc=com
# olcrootDN is the identity that is the root user for the LDAP database
olcrootDN:     cn=Manager,dc=example,dc=com
# olcrootPW is the password for the root user of the LDAP database
olcrootPW:     secret
olcdbhhost:    localhost
olcdbhport:    3306
olcdbhconnect: localhost
# olcdbuser is the identity that is the root user of the MySQL Cluster
database
#                (used to authenticate to the cluster).
olcdbuser:     root
olcdbpass:     secret
olcdbname:     OpenLDAP
# olcdbconnections comment: (you need one free [mysqld] in
#                config.ini for each dbconnection
olcdbconnections: 1
olcldbsocket:  /tmp/mysql.sock                # for localhost use
```

4 Converting an OpenLDAP Database from HDB/BDB to the Driver for MySQL Cluster Carrier-Grade Edition

This is a simple matter for administrators familiar with OpenLDAP and MySQL Cluster. The OpenLDAP people merely need to modify the database configuration for the database to use the MySQL Cluster (`back-ndb`) database engine. The rest of the OpenLDAP technology is unaware of the change. The MySQL Cluster administrators need to set up the data nodes to store the planned database.

OpenLDAP performs much of the detailed database initialization. OpenLDAP will automatically set up the cluster based on the configured LDAP schema. If you have an LDAP schema that accurately describes your existing HLR/HSS data then it is possible to re-use that schema. However, since OpenLDAP requires the addition of its own primary key column to every table that it uses, it is unlikely that you can use the existing data directly. At the very least it would need to be exported and re-imported.

The amount of hard drive space required for an OpenLDAP database supported by `back-ndb` will be similar to the space required for the same database under `back-hdb`. At present, `back-ndb` does not support substring indexing so `back-hdb` databases currently using substring indexing will take less space on `back-ndb`. This will likely be changed at some point in the future.

5 Further Considerations

There is one significant limitation in the current implementation of the OpenLDAP Driver for MySQL Cluster. At the moment, you can index a single combination of attributes. Filters can use the index if they match the index combination from left to right. E.g., if you index a, b, and c, then filters (a=foo), (&(a=foo)(b=bar)), and (&(a=foo)(b=bar)(c=baz)) can use the index. Any other order won't.

Obviously, it is very valuable to be able to use the index during search so it is important that the index be constructed to support the most frequent application queries.

Also, you will be able to access the LDAP data from MySQL using standard MySQL relational database commands. Assuming you set the database name to be OpenLDAP, the commands in Listing 3 will return a list of the tables used to store the LDAP data.

Listing 3 – Accessing OpenLDAP data via MySQL Cluster Commands

```
-mysql -uroot OpenLDAP  
-mysql> show tables;
```

The database name is entirely arbitrary and up to you or your enterprises naming conventions for MySQL databases. The use of OpenLDAP as the database name in this example is of no particular significance.

6 References

OpenLDAP: <http://www.openldap.org/>

Symas: <http://www.symas.com/>

MySQL Cluster on the web: <http://www.mysql.com/products/database/cluster/>

MySQL Cluster Datasheet: <http://www.mysql.com/products/database/cluster/mysql-cluster-datasheet.pdf>

MySQL Cluster Evaluation Guide: http://www.mysql.com/why-mysql/white-papers/mysql_cluster_eval_guide.php

7 About Symas

Symas™ Corporation was founded in 1999. The Founders originally set out to develop industry-leading and proprietary User Management software. The challenge of collecting, organizing, and auditing all the information about who has access to enterprise information technology is daunting. None of the offerings at the time offered practical solutions and the founders of Symas had an approach that offered unique advantages and real hope of tackling the challenge. This class of technology presents database challenges poorly addressed by Relational Data Base Management Systems (RDBMSs). These challenges were much more directly addressed by the features of Internet Standard LDAP directory data base management software.

In 1999 Symas elected to base its development on the then relatively young Open Source Software project, OpenLDAP. The project was working to prepare the University of Michigan's Open Source LDAP server software for broader deployment. It needed significant work on portability, architecture, and functionality. Starting from the beginning, Symas contributed continuously and heavily to the OpenLDAP project as a maintainer and developer.

With limited traction for its User Management efforts, the company evolved to survive the Dot-Bomb, doing custom programming, consulting, and continuing to provide enhancements and updates to OpenLDAP. Ultimately, the company focused all its efforts in building a commercial technical support, training, and consulting company around OpenLDAP. Today, Symas is committed to helping enterprises introduce new directory database applications for security, identity and network management and assisting them in converting existing directories to OpenLDAP.

As enterprise demand has increased, Symas has responded by increasing its support and strengthening its commercial OpenLDAP offerings. The result is Symas OpenLDAP, the leading

distribution of OpenLDAP and associated Open Source technologies. Companies like Yahoo!'s Zimbra unit, Sendmail, MDSI, Ventyx, Airwide, EMC, Sun, and Fidelity National Information Systems rely on Symas and Symas OpenLDAP for directory technology integrated into their offerings.

8 Acknowledgements

The authors wish to thank Sun Microsystems Inc. for providing much of the detailed information about MySQL Cluster Carrier Grade Edition and for permission to include it in this White Paper..

Copyright © 2009, Symas Corporation. Sun is a registered trademark of Sun Microsystems Inc. MySQL is a registered trademark of Sun Microsystems in the U.S. and in other countries. Other products mentioned may be trademarks of their companies.