Inventions on LDAP data management- a TRIZ based analysis

Umakant Mishra
Inventions on LDAP data Management
-A TRIZ based analysis

By- Umakant Mishra, Bangalore, India
http://umakantm.blogspot.in

Contents

Contents ........................................... ................................................... .................1
1. Introduction .................................... ................................................... ................2
2. Study on Inventions on LDAP data management ...........................................2
   2.1 Objectives of the study ................................................................. 2
   2.2 Major areas of Invention ............................................................... 3
   2.3 Patents analyzed for the study ....................................................... 3
3. The mechanics of LDAP data management .............................................3
4. Major Concerns in LDAP data management ...........................................4
5. IFR for LDAP data management ......................................................... 5
6. Inventions LDAP data management ..................................................... 6
   6.1 Bulk import to LDAP directory server ............................................. 6
   6.2 Online directory service with multiple databases ............................... 7
   6.3 Processing sparse hierarchical ACL data in relational database .......... 8
   6.4 Method of migrating data from one LDAP directory to another ......... 9
   6.5 Accessing LDAP data with a back-off capability ............................... 10
7. Other related inventions ....................................................................... 12
   7.1 Data replication for LDAP .................................................................. 12
   7.2 LDAP directory server cache mechanism ......................................... 12
   7.3 Scalable event notification in LDAP systems ..................................... 12
   7.4 Method of caching LDAP queries ..................................................... 13
   7.5 Trusted processing of Unique Identifiers in LDAP .............................. 13
   7.6 XML-LDAP Adapter for data transfer .............................................. 13
   7.7 Data management interoperability in heterogeneous directory structure .. 13
   7.8 Reverse indexing LDAP database for wildcard searching ................. 14
8. Summary and Findings of the study .................................................... 14
   8.1 Distribution of patents ..................................................................... 14
   8.2 Hot areas in LDAP data management .............................................. 14
   8.3 Trends of evolution in LDAP data management ................................ 15
   8.4 Predicting future inventions on LDAP data management ................. 15
Reference to patents: ................................................................................... 16
Other references: ......................................................................................... 17
1. Introduction

Lightweight Directory Access Protocol (LDAP) is an IETF open standard to provide directory services in the network. LDAP was initially developed at the University of Michigan with an objective to include most of the features of X.500, while eliminating the burdens and difficulties of the same.

The current Version of LDAP is LDAP V.3 released in December 1997 (RFC 2251). Other specifications of LDAP can be found in different RFCs on the IETF (Internet Engineering Task Force) website.

With the growing use of Internet, LDAP is becoming more and more popular to provide directory services to a wide range of applications. This led to patenting several inventions relating to LDAP operation and application.

This study on LDAP data management is a part of the main study on LDAP based on 60 selected patents on LDAP from US Patent database. (For more details on the study please refer to the article “Inventions on LDAP- A study based on US Patents”, by Umakant Mishra, cited in the reference.)

2. Study on Inventions on LDAP data management

This study on LDAP data management is a part of the above-mentioned study on LDAP. The study is mainly based on the analysis of US patents related to LDAP data management. Selection of patents and methodology followed in the study is presented in the article mentioned above.

2.1 Objectives of the study

- What are the inventions made on LDAP data management and on which aspects of data management?

- Which problems on LDAP data management are not solved yet? In other words which problems need to be addressed in future?

- Which aspects of LDAP data management are yet unexplored? What are the possible areas of improvements in future inventions?

- What is the IFR for LDAP data management? Is there any trend in the series of inventions?

- How is TRIZ useful in analyzing these inventions? Can we use Inventive Principles or any other TRIZ concepts to explain these inventions?
2.2 Major areas of Invention

- One major area that attracts inventors is to manage large LDAP directories especially in distributed environment. In such cases authentication, query management, referral services, speed of execution etc. become concerns.

- The other important area is the physical data management in terms of data import, export, conversion, migration, replication and other such operations on the LDAP database.

- The other major area of invention is to organize data to improve performance, storing in pre-fetched tables, using cache, organizing and updating cache etc.

- Besides there are inventions on managing special types of data (like ACL data) in RDBMS used as backing store for LDAP, issues like multiple concurrent access of data etc.

2.3 Patents analyzed for the study

The report presents a detailed analysis of following 5 patents on LDAP data management.


Besides the report also illustrates eight more patents, which are partially related to the topic of LDAP data management.

3. The mechanics of LDAP data management

The data model of LDAP is same as X.500 data model. The LDAP protocol assumes there are one or more servers, which jointly provide access to a
Directory Information Tree (DIT). The tree is made up of entries. Entries have relative distinguished name (RDN), which must be unique among all its siblings. The concatenation of the relative distinguished names from a particular entry to an immediate subordinate of the root of the tree forms the Distinguished Name (DN) for that entry. (IETF, RFC 2251).

![Functional Model of LDAP](image)

The LDAP server stores the directory information in a database. But LDAP does not provide any specification on the data storage. Different vendors can implement different mechanism for data storage as found convenient.

LDAP provides add, delete and modify operations for data modification. Each of these LDAP update operation is atomic. That means the whole operation is processed as a single unit of work. If a modify request is supposed to affect multiple attributes within an entry, it has to either affect all the attributes or none of the attributes.

LDAP provides the capability for directory information to be queried or updated. It offers a rich set of searching capabilities. The client makes a TCP/IP connection and sends requests to an LDAP server. The Server executes the query, and returns a sequence of responses to the client. The response may contain entries found during the search or referral to other servers.

According to LDAP protocol specifications an LDAP server may hold cache or shadow copies of entries, which can be used to answer search and comparison queries. But it does not mention any specific mechanism of caching.

**4. Major Concerns in LDAP data management**

- As LDAP does not specify any specific mechanism for data storage, different vendors implement different storage mechanism, which causes incompatibility in certain operations like directory merging or integration.

- The hierarchical LDAP data does not suite to be stored and processed in a relational database (RDBMS). But RDBMS is a proven database mechanism and desirable to be used for LDAP backing store.
- The database schema can be different for different directories, which causes problems in interchanging data between directories.

- Although LDAP supports caching, it does not specify any specific caching mechanism. Different vendors have to implement their own caching mechanism.

- Updating data by multiple concurrent edit-requests may cause some problem in data management and limitations in serving to queries.

- Although LDAP supports referral to other servers, but the actual implementation of clients’ queries to referral servers is left to the vendors.

5. IFR for LDAP data management

According to TRIZ, ideality is a function of its benefits and costs. $\text{Ideality} = \frac{\sum \text{Benefits}}{\left( \sum \text{Costs} + \sum \text{Harm} \right)}$. So the Ideal Final Result in LDAP data management can be achieved by increasing all useful functions of the data management and decreasing all harmful functions of the same. An analysis of the LDAP system may find the following as IFR for LDAP data management.

<table>
<thead>
<tr>
<th>Should have (Positive features)</th>
<th>Should not have (Negative features)</th>
</tr>
</thead>
<tbody>
<tr>
<td>• The LDAP server should be able to manage large number (virtually unlimited) of entries.</td>
<td>• Increasing the size of LDAP data should not affect its performance.</td>
</tr>
<tr>
<td>• The LDAP server should recognize other database formats and support import and export of data from other directories/databases.</td>
<td></td>
</tr>
<tr>
<td>• The LDAP server should manage data replication and data synchronization in a distributed environment.</td>
<td>• Should not demand excessive bandwidth or system resources for such operations.</td>
</tr>
<tr>
<td>• Should allow concurrent data searching and updating in distributed environments.</td>
<td></td>
</tr>
<tr>
<td>• The LDAP should protect its data from unauthorized access (may be implementing password securities, user policies, encryption and other such things) and maintain security and confidentiality.</td>
<td>• The security mechanism should not increase complicacy of the system.</td>
</tr>
<tr>
<td>• Should manage a cache of queries and responses for addressing repetitive queries.</td>
<td></td>
</tr>
</tbody>
</table>
The LDAP server should automatically manage its data, such as, maintaining data consistency, reliability, data backup and recovery, error detection, intrusion prevention etc.

The Ideal Final Results (IFR) are important as all the inventions try to achieve at least some of them fully or partially. However, practically it is not so easy to achieve IFRs as solving any of the above may lead to other problems and cause contradictions. For example, updating data in a distributed environment may lead to increased need of bandwidth and system resources for data synchronization. Inventors have tried to eliminate these contradictions and found many inventions to improve the LDAP data management system.

6. Inventions LDAP data management

As we discussed above, inventions are found on various aspects of LDAP data management. Some of the selected inventions are analyzed in the following pages.

6.1 Bulk import to LDAP directory server

Background problem

The LDAP protocol is a message-oriented protocol. The client constructs an LDAP message containing a request and sends the message to the server. The server processes the request and sends the result back to the client as a series of LDAP messages. In order to add an entry into the LDAP database, a client first opens a TCP connection to the LDAP server and submits the credentials for authentication such as passwords, digital certificates and similar. After the client is authenticated, the client issues a request to add or modify the data. This process is slow for adding a series of records, as the whole cycle of operation has to take place for each record to be added. There is a need for an improved method for importing large number of records into an LDAP database.

Solution provided by patent 6877026

US Patent 6877026 discloses a method of bulk import into LDAP server. According to the invention, the client will first send a request for an extended operation to the server. Then the client will perform the extended operation after receiving the extended operation request from the server. The client will send the object identifier of the extended operation and perform a series of LDAP add operations framed by extended operations to import an entry into the server.
The invention illustrates iPlanet Directory Server (iDS) that uses LDAP to provide centralized directory service. During bulk import, all LDAP operations within the iDS are suspended except the operations necessary to accomplish the bulk import. The bulk import feature of the iDS imports data using several different methods, such as, a Fast Replica Initialization method, a Wire Import method and a Direct Transfer method. This invention on bulk import provides increased reliability and performance of the directory service and can be used to import data to populate directories, merge directories, perform replication activities etc.

TRIZ based analysis

The bulk import executes faster because it eliminates unwanted activities like authentication, binding etc. for executing every add operation (Principle-21: Skipping).

During bulk import, all LDAP operations within the Server are suspended except the operations necessary to accomplish the bulk import (Principle-20: Continuity of useful action).

The invention describes several different methods of bulk import, such as, a Fast Replica Initialization method, Wire Import method and Direct Transfer method, each having advantages in different situations (Principle-3: Local Quality).

6.2 Online directory service with multiple databases

Background problem

Many organizations maintain directories of users that can be accessed by other users. In such cases the data is typically maintained on the Internet and accessed through web based forms. The user submits a query through an input screen and the result is searched from the databases. This method works fine when there is one database. But when there are several databases and each having large number of records, the performance deteriorates. There is a need to improve the data management in such scenario.

Solution provided by patent 5918227

US patent 5918227 discloses a method of accessing multiple directory databases though a “gater” or intermediate computer. As per the invention there is a first processor or “gater” in between the client and the directory servers. The
first processor receives requests and passes them to one of the second processors called “gaters”. The other “gaters” receive requests passed from the first “gater” but afterwards communicate directly with the user independent of the first “gater”. The “gaters” can communicate with each other to retrieve records. The invention gives an example of Switchboard directory service (and not LDAP).

**TRIZ based analysis**

Data searching in online directory should take no time (or minimum time) even with multiple databases each having large number of records (Ideal Final Result).

The client sends request through the first directory server (“gater”) instead of sending queries to each directory servers (Principle-24: Intermediary).

Although other “gaters” get request through the first “gater”, they communicate directly with user independent of the first “gater” (Principle-21: Skipping).

**6.3 Processing sparse hierarchical ACL data in relational database**

**Background problem**

An Account Control List (ACL) contains the information on what distinguished names (DN) have permissions to perform particular actions on an entry. For the purpose of security, this ACL data is stored in the database along with the data it protects. This ACL data is a sparse hierarchical data. In past, such hierarchical data was stored in an editable flat file as it is difficult to manage within a relational database. But in such cases where LDAP uses a relational database as its backing store, it is necessary to find some mechanism of storing and processing ACL data in relational database.

**Solution provided by patent 6823338**

US Patent 6823338 discloses a method for processing Access Control List (ACL) for LDAP in a relational database. This method is similar to the invention in patent 6438549 (by the same inventors) mentioned elsewhere in the study, which stores ACL data in multiple tables, such as an owner table, a propagation table, a permissions table and a source table (Principle-1: Segmentation). According to the
Inventions on LDAP data management, by Umakant Mishra

invention, the first three tables are used to determine the entry owner and ACL for a given object. Whereas the fourth table (the source table) is used to significantly increase the searching speed.

When a SELECT operation is performed based on an identifier of the object, if an ACL or owner is found, that value is kept. If, however, either the ACL or owner is not determined, the parent is checked. If the needed value is then found, a propagation flag is checked. If the propagation flag is TRUE, that value is kept. If the propagation flag is FALSE, then processing continues recursively until both an owner and an ACL value have been found. If the top of the tree (the suffix) has been reached without locating a propagating value, then system defaults are returned.

According to the invention the fourth table, i.e., the source table keeps track of the actual entry that holds the owner information and the entry that holds the ACL information for each object. This greatly increases the performance for search operations by finding the ACL and owner information for a particular object by only a single SQL call. Without this table a directory search would require the ACL and owner information to be retrieved for each object using multiple SQL statements.

**TRIZ based analysis**

The ACL data is stored in multiple tables such as an owner table, a propagation table, a permissions table and a source table (Principle-1: Segmentation) for convenience in storage and processing.

While the first three tables keep the ACL data, the fourth table keeps a list of pre-searched data for reference which increases speed of searching (Principle-10: Prior Action).

**6.4 Method of migrating data from one LDAP directory to another**

**Background problem:**

It is often required to migrate (e.g., copy) data from one LDAP database to another. Migrating data from one database to another database is not problematic if both databases have the same schema. However, if the databases do not have the same schema, then problems arise because of incompatibility.

**Solution provided by US Patent 6915287:**

US Patent 6915287 discloses a method and software tool that is useful when one is attempting to migrate data from LDAP directory to another LDAP directory where the schema for the two directories are not the same. The tool first compares the schema of a source directory to the schema of the destination directory. Then it updates the schema of the destination directory to be compatible with the source directory’s schema.
Before adding the entries, it verifies the modified schema with each entry from the source database to check whether it can allow the entry to be added to the directory. In case the destination directory’s schema does not allow any particular entry, the destination directory’s schema is modified again until it ensures that the schema is fully compatible. Finally it adds the entries to the destination directory.

TRIZ based analysis:
The method first compares the schema of the source directory to that of the destination directory and updates the schema of the destination directory to make it compatible to the schema of source directory (Principle-33: Homogeneity).

Before adding the entries, the method verifies the modified schema to ensure that it can take all entries from the source database (Principle-10: Prior Action).

In case the destination directory’s schema does not allow any particular entry, the destination directory’s schema is modified again and again until it ensures that the schema is fully compatible (Principle-20: Continuity of useful action).

6.5 Accessing LDAP data with a back-off capability

Background problem:
LDAP protocol allows LDAP clients to update their information such as name, address, email etc. on the LDAP server. It is possible that an LDAP client access the LDAP server to alter the policy definitions at any time. Since LDAP protocol has no locking mechanism, a variety of problems may result when some clients attempt to access the LDAP server for information while another client is in the process of altering the information contained in the LDAP server.
Desired result

It is necessary to find a mechanism to enable LDAP clients to read information from an LDAP server directory while another LDAP client is attempting to update information.

Solution provided by patent 6622170

US Patent 6622170 provides a method for LDAP client database access with backoff capability. The invention maintains a current tree of the directory information, which is never altered. Whenever an LDAP client wishes to update configuration information (i.e., add, change or delete), a new tree is created by cloning the current or a previous tree or by building a new tree. When the LDAP client is finished updating the new tree, the path for using LDAP clients is set to the new tree and the clients are requested to read LDAP policy configuration information using the new path. If the new directory tree of policy configuration information is found to be defective, the path definition of the information retrieval LDAP client is reset to the original information tree and the information retrieval LDAP client is requested to re-read the LDAP information, in this case, the original policies.

TRIZ based analysis:

The current tree of information, which is being used by the clients to retrieve directory information, is never altered. When an LDAP client wishes to update directory information, a copy of the directory is created (Principle-26: Copying).

Instead of transferring the new (updated) directory tree to the original path, the path definition of the LDAP client is set to the new (updated) directory (Principle-13: Other way round).

If for some reason the new directory tree is found to be defective, the path definition of the LDAP client is reset to the original information tree (Principle-11: Cushioning).
7. Other related inventions

7.1 Data replication for LDAP

The existing methods of data replication are not efficient to update LDAP data in a distributed environment where the replication sites may have different schema of databases. There is a need for a method of data replication in heterogeneous environments.

US Patent 6615223 discloses a method of data replication for LDAP servers even in heterogeneous environment. According to the invention, when the first server will receive a change request it will change the data in that site. Then it will generate a change record in a schema independent format (Principle-6: Universality) and send that change record to the second replication site. The second replication site will understand the change record as it is in schema independent format and will implement the corresponding changes in that site.

7.2 LDAP directory server cache mechanism

The LDAP server executes clients’ queries and sends the responses back to the LDAP clients. In practice there are lot of repetitive queries to the LDAP, which takes up a lot of time to execute the same queries again and again.

US Patent 6347312 discloses a method of faster searching of LDAP data in RDBMS. The invention uses a cache to store the search results retrieved in response to the search queries (Principle-27: Cheap and disposable). The invention prefers to use two sets of caches (Principle-1: Segmentation). The first cache (named Type-I cache) receives a set of identifiers indexed by a filter key of the search query. The second cache (named Type-II cache) stores the search results or the entries corresponding to the set of identifiers.

When the same queries come again and again, the LDAP server serves the results from the cache (Principle-3: Local Quality) instead of sending requests to the backend database. When data is updated in the directory data the corresponding cache is invalidated (Principle-34: Discard and recover) to maintain integrity of the cached information.

7.3 Scalable event notification in LDAP systems

A persistent search is an ongoing search that provides a mechanism by which an LDAP client can receive notification of modifications in an LDAP database. Each persistent search maintains an open TCP/IP connection to the LDAP server, which affects the performance of LDAP server.
US Patent 6769011 describes a method of more efficient, less resource-intensive and faster directory searches in LDAP servers. According to the invention, a portion of the directory server is searched using a proxy, and the proxy also gets registrations from the clients (Principle-24: Intermediary). The proxy can combine registrations from multiple clients into a single search of the LDAP directory (Principle-40: Composite). Registrations by the clients can be restricted to the portion of the LDAP directories searched by the proxy. The LDAP servers notify the proxy concerning modifications to entries in the LDAP directories, which the proxy in turn notifies to the appropriate clients.

7.4 Method of caching LDAP queries

US Patent 7035846 provides a method for processing directory queries in LDAP through a proxy server. The proxy server is configured to receive the directory queries from the clients. The proxy server maintains a history of queries and results in the cache (Principle-10: Prior Action). It compares the query characteristics stored in the cache and determines whether the query can be answered from the cached queries. If the query (or subsets of the query) is found to be there in the proxy cache, the proxy server retrieves the data from the cache (Principle-3: Local Quality) and responds to the client, otherwise it sends a request to the LDAP server.

7.5 Trusted processing of Unique Identifiers in LDAP

US Patent 6714930 presents a method, which allows different security systems to store and retrieve unique identifiers that are shared or common to the entire directory. According to the invention, there will be trusted process, which will generate or verify a unique identifier, and guarantee the uniqueness of the unique identifier within the entire directory (rather than just within a single security system), and allows all the security systems to share the unique identifier information (Principle-6: Universality).

7.6 XML-LDAP Adapter for data transfer

US Patent 7076488 discloses an arrangement for transforming data between an XML (Extensible Markup Language) data source and an LDAP interface. The arrangement includes an XML-LDAP adapter for transforming first data into second data (Principle-36: Conversion).

7.7 Data management interoperability in heterogeneous directory structure

Patent 6484177 discloses a data management system, which enables different heterogeneous systems to exchange information. The systems may be comprised of one or more networks, simple or complex file system, different types of databases or directory services like LDAP or Microsoft Active directory, they can still communicate with each other. The invention discloses a group of data managers and a user interface through an API (Principle-24: Intermediary).
7.8 Reverse indexing LDAP database for wildcard searching

US Patent 6199062 discloses a method of wildcard searching in relational database used for LDAP data storage. The relational database normally includes a forward index of the character strings in the database. The new method generates a reverse index of the character strings in the relational database that improves performance in wildcard searching (Principle- Other way round). When there is no wildcard the method uses the forward index and when there is a wildcard searching the method uses the reverse index (Principle-15: Dynamize).

8. Summary and Findings of the study

8.1 Distribution of patents

- A total of thirteen patents are illustrated in the study. Five of them are more related to LDAP data management and are analyzed in detail. The rest of the patents are analyzed in brief.

- Out of the four patents analyzed in detail, IBM assigns two patents, Sun Microsystems, Switchboard Inc and Novell Inc. assign one patent each.

- There are a large number of patents on data management in general (and not specifically for LDAP data) which are not included in this study.

8.2 Hot areas in LDAP data management

The analysis finds that the inventions try to improve the following aspects of LDAP system.

- Managing multiple LDAP databases in a distributed environment.

- Storing and processing ACL data in RDBMS for LDAP.

- Managing unstructured, semi-structured and differently structured data.

- The physical data management including import, export, conversion, migration and replication of LDAP data.

- Accessing multiple LDAP directories with a single query.

- Concurrent multiple access in update operations.

- Maintaining and processing LDAP cache, serving queries from cache.
8.3 Trends of evolution in LDAP data management

The following trends are prominent in the inventions on LDAP data management.

- Increasing speed in updating- Inventions want to increase speed of editing and updating data especially in large distributed directories.

- Increasing segmentation- the LDAP database is spread across multiple servers to facilitate faster access for scattered geographical locations.

- Enterprise Integration- although the data is spread across multiple directory servers in different formats and schemas, inventors try to integrate them all for direct centralized access.

- Increasing controllability- Inventions try to increase the reliability of database operations such as concurrent modifications, replications and synchronizations.

- Increasing easiness to “configure and implement” and increasing easiness to “maintain and administer” are two obvious trends.

- Increasing standardization- Inventions are looking for Increasing interoperability to facilitate integration with other systems, databases or services, supporting conversion and migration.

- Increasing reusability- Inventions try to eliminate repetitive data entry. The data should be entered or updated only once but the changes should be reflected in all different directories using that data.

8.4 Predicting future inventions on LDAP data management

Based on the analysis we can find a lot of issues, which are still open to be solved in the field of LDAP data management. We can predict to see more inventions on the following aspects of LDAP data management in future days.


- Failure management of LDAP servers (Principle-11: Cushioning).

- Improving the structure and management of LDAP cache (Principle-38: Improve Quality).


- Validating data submitted to LDAP server (Principle- Prior Counteraction).


Reference to patents:


Other references:


About the author

After working for more than 18 years in various fields of Information Technology Umakant is currently doing independent research on TRIZ and IT since 2004. He last worked as Director and Chief Technology Officer (2000-2004) in CREAX Information Technologies (Bangalore). Before that he worked as IS/IT manager (1996-2000) for ActionAid India (Bangalore).

Umakant is a Master in Philosophy (MA), Master in Business Administration (MBA), Bachelor in Law and Logic (LLB), Microsoft Certified Systems Engineer (MCSE+I), Certified Novell Engineer (CNE), Master Certified Novell Engineer (MCNE), Certified Intranet Manager (CIM), Certified Internet Professional (CIP), Certified Software Test Manager (CSTM) and holds many other global IT certifications.