Global directory over LDAP with index servers

Petr Matoušek, 1. ročník PDS, ÚIVT FEI
školitel: Doc.ing.Miroslav Švéda

Abstract

This article discusses a problem of implementation of global directory over LDAP protocol. We tested both models which are proposed for LDAP - referrals and replication. However, there is still a major issue - uncontrolled search of directory name space. We believe that indexing could help to solve this issue. This work brings brief ideas and shows a direction of our future work.

1 Introduction

A directory service replies to requests to search for a user's e-mail addresses, phone numbers etc. Current directory servers are based on a new Internet protocol LDAP. This protocol is derived from X.500 design. It simplifies the X.500 model by eliminating list and read operations, uses a simple encoding schema and runs over TCP/IP instead of full ISO/OSI stack.

Many software producers support the LDAP protocol in their products. Although the LDAP protocol is platform independent and designed to be distributed there are many constraints to build a real global directory. Data can be shared using *referrals* or *replication.*

2 Referrals versus replication

There are two approaches how to share data in a global directory:

- data replication from one site to another
- referral (pointers) from one server to another at DIT (Directory Information Tree).

Referral is a pointer to another LDAP server specified in the local LDAP server configuration. Instead of storing the global database in one place a directory can be distributed to many places with pointers to servers which holds one or several parts of a shared database. Using referrals we can avoid high-volume database replication and data are still up-to-date.

The LDAP protocol does not implement chaining - processing referrals by servers. A client should process them instead. Unfortunately many LDAP clients are not matured to follow referrals correctly. If they follow a referral there is a danger of bouncing between servers. Following referrals is time unpredictible even if the time limit on the server side is defined.

Data sharing via **replication** suffers from incompatibility of LDAP servers from different producers. Some servers require additional attributes to establish a master-slave connection - a schema of slave directory, a set of slave's ACLs etc. The idea of a global directory based on an independent internet protocol is lost and only systems of the same producent can use replication principles for data sharing.

We tested both approaches but we did not succeed to interconnect differrent local LDAP servers into a global directory. The major problem was server incompatibility (replication) and client imaturity(refferal processing).

3 Indexing

Further analysis showed that even if both clients and servers process the LDAP requests properly **we cannot build a global scalable directory over LDAP** - if a request is not found on the local server it is propagated to every referred server. Concequential searching of wide directory name space cannot be controlled - the same problem that the X.500 protocol had as well.

We studied the behaviour of other distributed systems (DNS, a system of shared memory on distributed processors, whois++) and we propose an **index server** as a good solution to optimize and speedup LDAP searching. In our proposal the index server holds a list of referrals to DNs (Distinguished Names) of all interconnected LDAP servers. If an entry is not found in a local server a request is passed to the index server. The index server searches its database and returns a set of referrals. The referrals are processed by the requested LDAP server and the result is returned to the client.

This system can be implemented over LDAP protocol. However, many operations concerning index server communication would be slower. So we prefer to extend the definition of the LDAP protocol for index server operations.

4 Conclusion

This paper showed major issues which occurs implementation of global directory. We proposed a model how to solve the problem of the uncontrolled name space searching. We believe that index server can solve this problem. In our future work we intend to implement our ideas in LDAP servers.

References

[1] M.Wahl, T.Howes, S.Kille: *Lightweight Directory Access Protocol (v3)*, RFC 2251, 1997.

[2] *ITU-T Rec. X.501 - The Directory: Models*, 1993.

[3] *The SLAPD and SLURPD Administrator's Guide*, University of Michigan, 1996.

[4] P.Deutch,R.Schoultz,P.Faltstrom,C.Weider: *Architecture of the Whois++*, RFC 1835,Bunyip Information System, Inc., 1995.

[5] P.Faltstrom,R.Schoultz,C.Weider: *How to Interact with a Whois++ Mesh*, RFC 1914, 1996.