Usage of LDAP in Globus

Gregor von Laszewski and Ian Foster
Mathematics and Computer Science Division
Argonne National Laboratory, Argonne, IL 60439

gregor@mcs.anl.gov

Abstract:

This short note describes the use of LDAP in the Globus metacomputing toolkit. It answers three questions: What is LDAP? Where is it used? and Why is it used in the Globus metacomputing toolkit?

Contents

1. Short Introduction to Directory Services and LDAP

To describe the role of LDAP in the Globus metacomputing toolkit, we must first define the necessary terminology. It is important to understand what a *directory*, a *directory service*, and *LDAP* are.

Directory:

Directories are used to store and retrieve information. Thus, directories are similar to databases. Special characteristics of directories include following:

- Directories are designed for reading more than for writing.
- Directories offer a static view of the data.
- Updates in directories are simple without transactions.

Directory Service:

A Directory Service provides a directory that can be accessed via a network protocol. Often, directory services include mechanisms for replication and data distribution.

An example of a directory service is Domain Name System (DNS), which resolves the names of computers to appropriate addresses. Programs using this service are countless: for example, finger and mail.

LDAP:

The abbreviation LDAP stands for *lightweight directory access protocol*. LDAP defines a standard directory protocol that includes the following features:

- A network protocol for accessing information in the directory.
- An information model defining the form and character of the information.
- A namespace defining how information is referenced and organized.
- An emerging distributed operation model defining how data may be distributed and referenced (LDAP version 3).
- An extensible protocol.
- An extensible information model.

The information model and namespace are based on entries. An entry is used to store attributes. An attribute has an associated type and can have one or more values. Each entry in the namespace has a distinguished name that allows easy identification. The access of the data in an LDAP-based directory is accomplished by using the following:

base DN:

which indicates where in the hierarchy to begin the search.

filter:

which specifies attribute types, assertion values, and matching criteria.

scope:

which indicates how many levels of the directory tree are searched, relative to the base DN.

LDAP is a vendor- and platform-independent, open, network *protocol* standard. Hence, it can be ported transparently across a highly heterogeneous network, as required in metacomputing systems. Moreover, it is possible to write gateways between LDAP and other protocols or systems, which result in a highly accessible directory service. Current gateways include

- LDAP to X.500 and X.500 to LDAP,
- HTTP to LDAP,
- WHOIS++ to LDAP,
- FINGER to LDAP,
- Email to LDAP,
- ODBC to LDAP (in progress), and
- MDS to LDAP (see Section 2).

Beneficial for widespread use is the fact that LDAP-based directory supports any type of data. Furthermore, the LDAP protocol supports various forms of strong security (authentication, privacy, and integrity) technology.

2. Directory Service for a Metacomputing Infrastructure

For the implementation of a networked supercomputing environment or a metacomputer, information about the hardware, software, and status of the system has to be stored. This information can be used to implement a resource-aware infrastructure or metasystem.

The information service associated with a metacomputing environment must have the functionality of a *white pages directory* and *yellow pages directory*. Their functionality is best described with a simple example:

White Pages:

look up the IP number, amount of memory, and so forth, associated with a particular machine.

Yellow Pages:

List all the computers of a particular class or with a particular property.

LDAP is used in the Globus metacomputing toolkit for tasks where yellow- and white-pages services are needed. These services are enabled through the directory service associated with the Globus metacomputing. The directory service is named *Metacomputing Directory Service*, or MDS. Currently, MDS answers questions related to compute resources and network resources.

The use of LDAP in MDS is divided into three categories:

1.

Initialization: using an information schema that describes the attributes associated with the different entry classes.

2.

Population: a framework to populate the directory with information.

3.

Querying: a framework to request information from the directory service.

Next, we describe the usage of LDAP for each of these categories.

2.1 Initialization

Part of the MDS is a specialized set of object definitions. The objects contain attributes that are specifically useful for network computing environments. We have designed a naming scheme that allows one to specify the properties not only of computers (compute resources), but also of the networks and the connections between compute resources. Tools are provided that make defining and modifying the information description scheme possible. The current object model is available through the World Wide Web at http://www.gllobus.org. The naming scheme is described in http://www.globus.org/mds/OCBrowser/globus_object_defs.html.

2.2 Population

The population of the MDS is based on a set of simple UNIX shell scripts. These shell scripts are modular and are executed with the familiar pipeline concept. For each object that is defined by the object model, a UNIX script is available to populate the object with information. This modular approach is important to grant easy portability between platforms. After gathering the information with these scripts, the information associated with two objects is stored as entries in MDS. Updates to the MDS should be generally done via the *mds-update* script.

2.3 Querying

Obtaining information from the MDS is defined via the LDAP protocol standard.

Since the MDS is based on LDAP, it is possible to use a variety of application interfaces to communicate with the actual LDAP server. This software independence is one of the important features that make LDAP an ideal candidate for a very diverse compute environment as used in Globus.

Thus, querying and updating information are easy while choosing the desired API for interfacing with the LDAP server.

Obtaining information from the MDS is defined via the LDAP protocol standard.

Since the MDS is based on LDAP, it is possible to use a variety of application interfaces to communicate with the actual LDAP server. This software independence is one of the important features that make LDAP an ideal candidate for a very diverse compute environment as used in Globus.

Thus, querying and updating information are easy while choosing the desired API for interfacing with the LDAP server.

2.4 Application Interfaces

The categories of LDAP enable one to use the application interface most appropriate to interface with the MDS and its LDAP server. Interfacing can be done via LDAP calls directly or via some tools as provided in the MDS.

The application interfaces available to access the MDS directly are available in

sh:

shell scripts from the University of Michigan and Netscape.

Perl:

perldap:

p erldap-0.02.tar.gz

LDAP Perl:

is a LDAP PERL package from netscape.

Net::LDAPapi:

Net::LDAPapi is a perl5 module a description and a readme file, as well as the source (Net-LDAPap i-1_21_tar.gz)can be found in Perl Oasis

Java:

Java APIs are available from Netscape and Sun. Sun provides the JNDI class that will be part of the next Java release.

C:

A C API can be found at University of Michigan and Netscape.

Because of the high availability of the API in diverse computer languages, one can choose the API and language most appropriate for the problem to be solved. More information about APIs can be found in FAQ.0

3. Globus Components That Use LDAP/MDS

We list the components of Globus that use LDAP currently.

MDS:

From the preceding section it is clear that the MDS is based on LDAP. Nevertheless, the design of MDS makes it possible to use other databases or directory services to provide the same functionality as currently provided via LDAP. MDS has support routines for initializing, updating, and querying the LDAP server, making it a specialized directory server.

The initialization and update functions are woven into some Globus toolkit components in order to simplify maintenance and usage of the heterogeneous Globus testbed. An example of the use of the MDS can be found at http://www.globus.org/gustotestbed.

Information Browsers:

To monitor the status of the testbed (GUSTO), several tools have been developed. These tools are available at http://www.globus.org/gustotestbed. The tools include the following:

- Current versions of the software deployed on the GUSTO testbed
- Object classes being used by the MDS

MDS contents browser for the (alpha)

and (beta) testbed.

- The current status of Globus resource managers installed on the testbed
- An interactive graphical display of GUSTO status
- Current status of GUSTO regression tests

Configuring Globus:

During the configuration of Globus the site has to register with Globus. This process includes the population of the MDS with the compute and network resources at the site. The information is gathered automatically by the MDS.

GRAM:

GRAM relies heavily on the information stored in the MDS. Special attributes are included in the MDS that describe the contact points of the gatekeeper. GRAM allows the uniform specification of a job request and the submission of the job to the underlying scheduler (LSF, ...).

GRUB:

GRUB assumes that the information about hosts, contact strings, and other information in the MDS are properly available upon initialization. The GRS is an example on how to do this. GRUB will soon be replaced by a new API called DUROC.

GRS:

The General Resource Selector is a graphical tool to design a resource description query for a particular application. Based on the information stored in the MDS, a high-level representation of the application and its requirements for the metacomputing infrastructure are formulated and submitted via GRUB requests.

topologyd:

Topologyd is a tool that monitors and predicts the status of the network connecting the compute resources in a metacomputing infrastructure. The information is forwarded to the MDS. The status of the topologyd can be access via a WWW page.

Nexus:

At present, Nexus uses a database based on information from the MDS. In the future, Nexus will directly communicate with the MDS.

MPI:

The MPI version of Nexus has the same requirements as CC++. Hostnames are looked up from the MDS.

C++:

The port of C++ based on Nexus needs to know the so-called contact point to communicate between hostnames. C++ uses the hostname in order to look up the contact point that is stored in the MDS. In this way each C++ program automatically configures itself, and some of the details of using Globus are hidden from the application user. C++ uses the Nexus communication library for interprocess communication. More information about C++ can be found at http://globus.isis.edu/ccpp.

HBM:

The HeartBeat Monitor is used to monitor the status of the compute resources in the testbed. We point out, that the HBM does **not** use the MDS. The intention is to minimize the dependency on other Globus toolkit components: we believe that the HBM should be independent. The status of the HBM can be accessed via a WWW page.

4. Conclusion

Many components of the Globus metacomputing toolkit make use of MDS, and we anticipate that this use of MDS will increase. Experiments are under way to explore different distributed LDAP servers. While it is possible to implement the functionality of MDS by using technologies other than LDAP, we feel that LDAP is a good fit and, with increasing acceptance of the Internet community, a state-of-the art choice. The high availability of application interfaces in different languages makes it possible to integrate LDAP queries into the toolkit components and application programs.

References

MDS / Globus

References to MDS and Globus can be found at http://www.globus.org.

LDAP

For more information on LDAP and Directories, refer to the following:

- http://www-leland.stanford.edu/ hodges/talks/
- LDAP and X.500 Roadmap, Jeff Hodges, periodically updated.
- This page provides a roadmap for learning about LDAP and X.500, along with links to many and varied documents and resources, including the standards document themselves.

  Stanford University Network Identifier Project (SUNet ID)

  - SUNet ID Requirements, RL "Bob" Morgan, 3/1/96
  - SUNet ID Design Proposal, RL "Bob" Morgan, 3/1/96
- SUNet Directory Services: Statement of Direction, Jeff Hodges, 11/24/96
- LDAP-based Routing of SMTP Messages: Approach at Stanford University, Jeff Hodges and Booker Bense, INTERNET-DRAFT, 3/26/97
- LDAP-based Routing of SMTP Messages: Approach Used by Netscape, Hans

Lachman, INTERNET-DRAFT, March 97

- DirConnect1, vendor-independent directory interoperability testing, sponsored by the Internet Mail Consortium, 1997.
- Vendor-specific directory information resources:

  These papers have salient information about directory services, though they are quite vendor specific.

  - An Internet Approach to Directories, Netscape, 1997.
  - Novell Directory Service (NDS) White Paper, Novell, 1997.
  - Exchange Directory Service (LDAP), Microsoft, 1997.
- LDAP: Programming Directory-Enabled Applications with Lightweight Directory Access Protocol, T. Howes and M. Smith, Macmillan Technical Publishing, 1997, ISBN 1-57870-000-0.

Appendix

The following Table lists a small set of example programs

contact hostname	Returns the contact string of the host	sh
manager	Returns the list of all available managers	sh
cpuload-all	Returns the cpuload of all compute resources	sh
cpuload	Returns the cpu load of tuva	sh
freenodes	Returns a list of managers. Managers with more than two free nodes get an asterix.	perl

Gregor von Laszewski 1998-04-21