Distributed Authentication in GRID5000

Sébastien Varette, Sébastien Georget, Johan Montagnat, Jean-Louis Roch, Franck Leprévost

Distributed Authentication in GRID5000

Sebastien Varrette[1,4], Sebastien Georget[2], Johan Montagnat[3], Jean-Louis Roch[4], Franck Leprevost[1]

[1] University of Luxembourg, CESI-LACS, Luxembourg
INRIA DREAM team [2] and CNRS I3S unit, RAINBOW team[3], Sophia Antipolis, France
[4] MOAIS/RAGTIME Project, ID-IMAG Laboratory, Grenoble, France

Abstract. Between high-performance clusters and grids appears an intermediate infrastructure called cluster grid that corresponds to the interconnection of clusters through the Internet. Cluster grids are not only dedicated to specific applications but should allow the users to execute programs of different natures. This kind of architecture also imposes additional constraints as the geographic extension raises availability and security issues. In this context, authentication is one of the key stone by providing access to the resources. Grid5000 is a french project based on a cluster grid topology. This article expounds and justifies the authentication system used in Grid5000. We first show the limits of classical approaches that are local files and NIS in such configurations. We then propose a scalable alternative based on the LDAP protocol allowing to meet the needs of cluster grids, either in terms of availability, security and performances. Finally, among the various applications that can be executed in the Grid5000 platform, we present μgrid, a minimal middleware used for medical data processing.

1 Introduction

This article is motivated by the need for a robust authentication system in the Grid5000 platform[1]. This French project aims at building an experimental Grid platform gathering at least 8 sites geographically distributed in France. The main purpose of this platform is to serve as an experimental testbed for research in *grid computing*. The researchers in each of the implied laboratories can deploy various applications on the grid, for instance for data analysis. Consequently, they will have to be able to authenticate on each of the connected nodes. The authentication system is therefore a key element in this project and more generally in the field of the cluster grid as it allows the allocation of resources. At least three constraints should be addressed by this system:

- *Availability*: the system should work even in case of punctual disconnections
- *Security*: privacy and integrity of the data should be assured.
- *Delegation*: each administrator of a site should be able to manage its own users.

This article is organized as follows: §2 precises the context by presenting a classification of computing grids and detailing the notion of cluster grid. This paper is mainly directed to Linux systems as they constitute major actors in the field of the grid computing[2]. Yet, our results can be extended to other systems. After introducing naming services (§2.2), a section will be dedicated to directories and the LDAP protocol (§3). A large part of this article is dedicated to experimentations (§4) where different configurations will be tested. The analysis of these experiments will lead to the proposition of an authentication architecture for the Grid5000 platform. Finally, §5 illustrates the running of the authentication system by presenting an application for medical data processing.

[1] `http://www.grid5000.org`

[2] IBM which is the most represented manufacturers in the list of the 500 most powerful machines in the world (`http://www.top500.org`) claims 161 Linux clusters in this list. That is three quarters of the IBM systems and near a third of all manufacturers.

2 Context

Computer grids, as defined in [1], are distributed infrastructures that gather thousands of computers geographically scattered and interconnected through the Internet. This type of platform that used to appear in the 90's knew several evolutions and can be classified today in two main families:

1. The *"Desktop Grids"* [2] typically steal idle cycles of desktop PCs and workstations through the Internet to compute parts of a huge problem. Whereas this type of grid has been recently integrated in the general problematics of computing grids [3], we prefer going on with separating both architectures.
2. The *"Computing Grids"* rather gather one or more dedicated clusters. A *cluster* connects several computers through a local network in order to provide a coherent set able to deal with parallel computations, network load balancing, fault-tolerance... Each machine is a *node* of the cluster. Of course, each user of the grid has to authenticate on each node. This can be done either by a local copy of the user's credentials on the nodes or by using a Naming Service (such as NIS[3]) able to broadcast this information across the network. In the case of grids, the naming service has a strong imperative of scalability, as will be exhibited further.

To be totally exhaustive, an additional distinction in the concept of computing grids is proposed here. This classification is based on administrative heterogeneity (as illustrated in Fig. 1) and allows to subdivide computing grids in three categories:

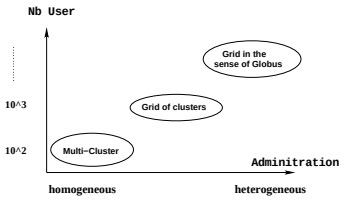

Fig. 1. Classification of computing grids based on administrative heterogeneity

1. Multi-clustering that bind together several nearby sites administrated by a single person that manages around 10^2 users.
2. Cluster grid that merge several scattered sites through the Internet. Each site is administrated by different persons, yet the administrative domains are sufficiently open to enable the settlement of conventions between the sites (for instance when resolving the hostname of the nodes, or for the choice of a common authentication system). Such a topology, illustrated in Fig. 2, manages around 10^3 users and corresponds to the architecture of Grid5000.
3. Finally, the grid in the sense of Globus [4] manages a huge number of users in sites administratively closed. Traditional authentication solutions proposed in this context can also be applied for the first two cases but are unadapted for these topologies. This article proposes an authentication architecture adapted for the three cases.

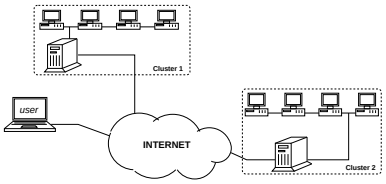

Fig. 2. A cluster's grid

[3] Network Information System

2.1 Authentication of users in Linux Environments

Under Linux, users authentication is based on two components:

1. The PAM system (Pluggable Authentication Module) which allows the transparent integration of various authentication technologies (as the UNIX standard based on `crypt`, RSA, DCE, LDAP) to be used by the services of the system as `ssh`, `passwd`, `su`, `ftp` ... PAM supplies an API by which the requests for authentication are mapped on specific actions related to the technologies used.
2. The NSS system (Name Service Switch). Once the user is authenticated, many applications still need to reach the information relative to this user. This information is traditionally contained in tables supplied either by local texts files (`/etc/passwd`, `/etc/shadow`, `/etc/group`, etc.) or by a naming service (see §2.2). NSS uses a common API dedicated to naming services that provided an intermediate layer between an application and a set of naming services. The NSS system is then able to access to the information of a given table (as `passwd` or `shadow`) using different naming services.

2.2 Naming services

For a PC not connected to the Internet, user authentication is achieved through local tables (`passwd` and eventually `shadow`) stored in files. Similarly, the name resolution of hostnames into IP addresses uses the file `hosts`. For a cluster, administrators prefers to use a Naming Service[4] able to broadcast authentication information across the network to authorized nodes. The information is centralized on one or more servers, making the administrative task easier: without such service, every single node should maintain its own copy of the information.

NIS. Introduced by Sun in 1985, the Network Information Service is used to centralized administration of systems information. The information is stored in maps under indexed databases (db, dbm) reachable by RPC[5]. NIS uses a Master/Slave model but does not allow the treatment of important volumes of data (each modification involves the transfer of the totality of the base). Furthermore, it is particularly hard to organize the data in a hierarchical way and the access security remains weak. In spite of all these drawbacks, NIS remains a well used system at the level of clusters and local networks mainly because of its installation simplicity.

NIS+. It was the answer of SUN to the drawbacks of NIS.NIS+ introduces the distribution of the data between master and slave in an incremental way, by adding the notion of hierarchical tree for the data. The use of certificates solves the security issue. Yet, the luck of flexibility in the hierarchical structure together with a too complicated installation proceeding slow down the passage from NIS to NIS+. Nevertheless, this approach prefigures the concepts used in this article.

NetWare NDS. Among the various proprietary solutions available, NetWare[5] is a local-area network (LAN) operating system that runs on a variety of LANs.It provides users and programmers with a consistent interface that is independent of the actual hardware used to transmit messages. In particular, NetWare NDS is a version of the NetWare file server operating system. NDS stands for Netware Directory Services, and is a hierarchical directory used to manage user-IDs, groups, computer addresses, printers and other network objects in a convenient manner. NDS is then used to retrieve the information required by an authentication process. This approach is finally closed to the one presented in this paper but has the inconvenient of a prohibitive cost unadapted for our context.

[4] DNS (Domain Name Service), NIS and NIS+ for instance
[5] Remote Procedure Call

3 Directories and LDAP

A directory is like a database, but tends to contain more descriptive information. Directories are tuned to give quick-response to high-volume lookup or search operations. They may have the ability to replicate information widely in order to increase availability and reliability, while reducing response time.

Based on X.500 protocol (ISO norm for the management of electronic directories), LDAP [6] is a standard access protocol to electronic directories allowing to perform researches and modifications. LDAP is based on the model of DNS: data are naturally organized in a tree structure and each branch of the tree can easily be distributed among different servers. LDAP technology has been adopted by many large companies. The generalization of LDAP has also been implemented in the applicative bases (some operating systems, like Mac OS X or Solaris 9, integrate a LDAP directory). LDAP proposes mechanisms to manage authentication, authorization and confidentiality of the exchanges. Indeed, several methods of authentication corresponding to various security levels are available (login/password, login/password on SSL, X.509 certificate etc...). The possibilities of authentication can be extended with the SASL[6] API allowing to easily integrating new mechanisms of authentication like Kerberos The applications thus delegate the step of authentication to the LDAP directory which implements one of the quoted methods. To conclude in terms of security, LDAP provides various guarantees thanks to the integration of cypher and authentication standard mechanisms (SSL/TLS, SASL) coupled with Access Control Lists. All these mechanisms enable an efficient protection of transactions and access to the data incorporated in the LDAP directory. Thereafter, practical experimentations on LDAP were done through the open-source and reliable implementation OpenLDAP[7].

Organization of data in LDAP LDAP data are organized in a tree structure called Directory Information Tree (DIT). Each node of the tree corresponds to an entry of the directory. An example of DIT is provided in Fig 3. Each entry is

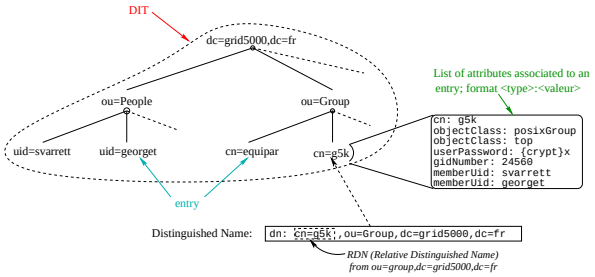

Fig. 3. Example of DIT : case of the users management

DC	domain components
OU	organizational unit
O	organization
CN	common name
SN	surname
UID	user ID

Fig. 4. Main abbreviations used in the DN field.

referred in a unique way by its *distinguished name (DN)*. This unicity is obtained by combination of the attributes listed in tab 4. The directory service provided by LDAP is based on a client/server model. The servers can be organized in the following configurations:

1. *Local Directory Service*: a unique server is able to deal with the clients requests.
2. *Local Directory Service with Referrals*: the server is configured to provide directory services for a local domain and to return referrals (i.e. a pointer) to a superior service capable of handling requests outside the local domain.

[6] Simple Authentication and Security Layer
[7] `http://www.openldap.org`

3. *Replicated Directory Service*: partial replication can be operated between master and slave servers.
4. *Distributed Local Directory Service*: the database is divided in subparts (eventually replicated) that are accessible through a set of referrals between the servers.

The last two modes will be particularly interested in our context.

LDAP vs Databases LDAP is often compared to a database. It is globally the case even if differences exist: see tab 1.

Criteria	**LDAP**	**Databases**
R/W ratio	read optimized	R/W
scalability	easy (LDAP schema)	hard
Table distribution	inherent	rare [7]
Replication	possible	possible
Transactional model	simple	advanced
Standard	yes	no (specific to SGBD)

Table 1. Advantages/Drawbacks of LDAP on Databases

LDAP vs. NIS This article compares authentication solutions based on LDAP to NIS. Tab 2 briefly introduce the characteristics of every system.

Criteria	**LDAP**	**NIS**
Port	specific (389/636 by default)	arbitrary (RPC)
Data privacy	possible	impossible
Access control mechanisms	yes	no
Table distribution	yes	no
Replication	yes (partial replication available)	yes (total repl. only)
Researches Semantics	advanced	simple

Table 2. Advantages/Drawbacks of LDAP on NIS

4 Experimentations

The comparison of authentication systems was realized on the client side by computing the number of simultaneous authentications the server can handle. So, the measures take into account the three elements of the identification chain: the PAM module, the transport layer between client and server and the delay in server's response. This approach allows to obtain results which correspond to the reality and not to the theoretical performances that the servers are supposed to achieve.

4.1 Local model

While being hard to maintain, the local duplication on each nodes of the system tables `passwd`, `shadow` and `group` is not the most effective solution (see fig.5).

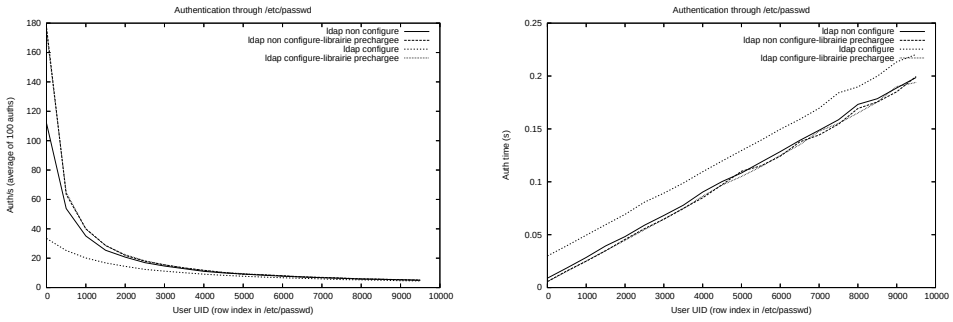

Fig. 5. Local table model: Impact of the location in the table on authentication time

During the authentication process, the files are sequentially read. The authentication time directly depends on the number of entries in the table. This is naturally unadapted to the case of clusters and grids (with high number of users). We also show the influence of preloading the libraries involved.

4.2 Comparison between NIS and the local model

Contrary to the local model, NIS ensures that the authentication time is globally independent from the number of entries in the table (see fig. 6). We know experiment different configuration of LDAP to justify our proposition for Grid5000.

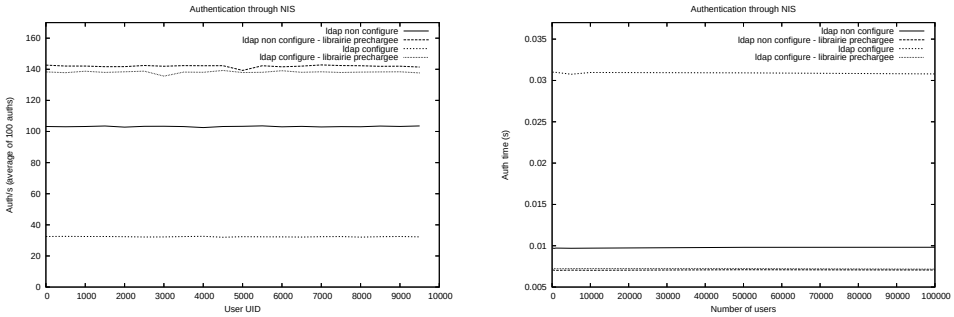

Fig. 6. In NIS, the authentication time is independent from the base size

4.3 Centralized client/server model

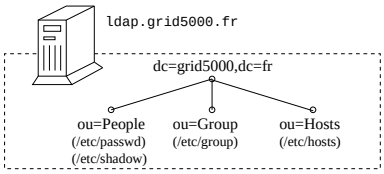

Fig. 7. Centralized client/server model

In this model, the LDAP server is configured in a similar way to a NIS server: the contents of all the tables are on the same server reachable by every nodes of the grid. Fig.7 illustrates the structure of the table in the LDAP server. Before comparing this solution with NIS, we wanted to estimate the impact of the configuration of the LDAP server LDAP on its performances. This is done in the following sections.

Impact of data indexing in LDAP configuration. When installing a LDAP server, a basic indexing is proposed but fig.8 shows that it *should not be used*. An appropriate indexing[8] should be preferred. This configuration guarantees constant performances with regards of the number of entries in LDAP base. Using SSL divides

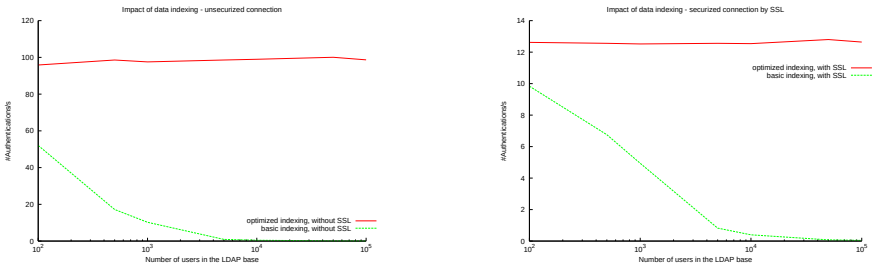

Fig. 8. Impact of data indexing on the LDAP server performances.

the performances by 9. This will be confirmed in §4.3. This is required for resources authentications and communication privacy.

Impact of the log level. The LDAP server can log different information represented by a level (see tab.9). They can be combined by addition. Fig.10 shows that each level has a different impact on the server performances.

[8] see `http://www.openldap.org/faq/data/cache/42.html` for instance.

Niv.	Description	Impact
-1	enable all debugging	++++
0	no debugging	-
1	trace function calls	+++
2	debug packet handling	0
4	heavy trace debugging	+
8	connection management	++
16	print out packets sent/received	0
32	search filter processing	+
64	configuration file processing	0
128	access control list processing	+++
256	stats log connections/op/results	+
512	stats log entries sent	~0
1024	print coms with shell backends	0
2048	print entry parsing debugging	0

Fig. 9. Log levels in OpenLDAP

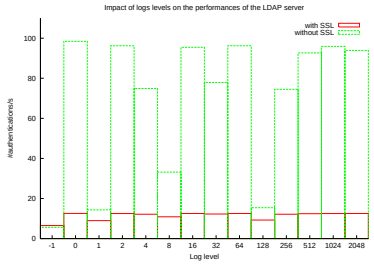

Fig. 10. Impact of logs levels on performances

Impact of the backend used OpenLDAP supports a variety of database backends which you can use ("`ldbm`" by default). The first measures made to compare the most used backends does not reflect real differences in the server performances.

Impact of SSL on LDAP As seen in §2.1, user authentication uses two components: PAM and NSS library. In LDAP, a specific interface must be used for each of these libraries. It can be configured to use SSL or not in the communications between the client and the server. As mentioned in §4.3, SSL divides the performances by 9 (see tab.3). This influence can be explained when analyzing the number of messages exchanged during an authentication by `ldap` or by `ldaps`. When using the `ldap` protocol, 45 LDAP messages are exchanged together with 35 TCP messages. With `ldaps`, it is 70 TLS messages and 47 TCP messages. Communications are thus more important, and encryption/decryption time should be taken into account.

PAM		NSS		#authentications/s		
ldap	ldaps	ldap	ldaps	Interval	Average	with preload
X		X		between 94 and 97	95.99	145.9
X			X	between 92 and 95	94.15	144.4
	X	X		between 12 and 13	12.46	14.28
	X		X	between 12 and 13	12.51	14.27

Table 3. Impact of using SSL in PAM and NSS libraries on the performances - intra-cluster Measures

Impact of inter-cluster latency The measures presented in the tab.4 were made with a server located in Sophia Antipolis while clients belongs to the cluster of Grenoble. Performances are divided by 4 because of latency and network disturbances. Consequently, the inter-clusters communications should be minimized in the authentication process.

PAM		NSS		#authentications/s	
ldap	ldaps	ldap	ldaps	interval	average
X		X		from 15 to 22	20.1
X			X	from 7 to 23	17.6
	X	X		from 6 to 7	6.89
	X		X	from 5 to 7	6.79

Table 4. Impact of inter-cluster latency

We realized a similar analyse with a NIS server. Results are displayed in tab.5 As communications are less important in NIS, the performances are globally better with it. Yet, the advantage of LDAP comes from its capacity to distribute the tables with eventually a partial duplication. This configuration is presented in the following section.

Latency type	#authentications/s			
	interval	average	with preload	average with preload
Intra-Cluster	from 230 to 310	263.0	from 266 to 338	290
Inter-Cluster	from 32 to 37	35.1	from 31 to 38	35.3

Table 5. Impact of latency when using a NIS server

4.4 Distributed "flat" model

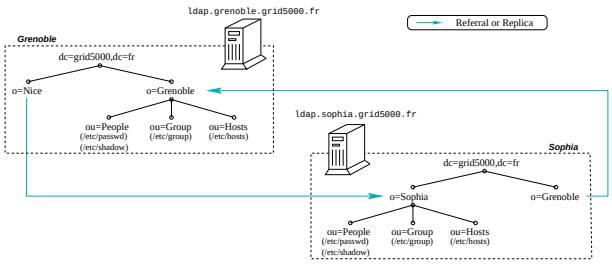

Fig. 11. flat distributed model for authentication based on LDAP

Based on a DNS model, LDAP allows the distribution of the tables on multiple servers. As before, the tree structure used to store the data in the LDAP server follows the organization of the sites in the grid but here, each site is responsible for a sub-tree containing data relative to the users and the resources of the site as illustrated in fig. 11. Reaching the data contained in other branches can be done in two ways:

- by using referrals as a pointer to a server able to answer a request
- by partial replication of some or all the other branch's.

We propose a partial replicated approach based on the following organization :

- each site is master for its branch and slave for the others
- A special backend (`meta`) aggregates each branch into the LDAP bases.

	backend `meta`		#auth./s (`ldap`)			#auth./s (`ldaps`)			
Mode	**Grenoble**	**Sophia**	**min.**	**avg.**	**max.**	**min.**	**avg.**	**max.**	**Comments**
centralized	inact.	inact.	84.9	89.0	107.3	16.7	17.4	17.9	Centralized mode (see §4.3)
replica	inact.	inact.	82.9	96.6	108.9	17.2	17.4	17.8	Impact of replica configuration
replica	local	local	82.0	87.9	97.1	16.9	17.1	17.5	User managed in Grenoble
replica	local	local	81.3	88.9	97.6	16.8	17.2	17.5	User managed in Sophia
replica	local	remote	35.8	36.8	38.3	13.3	13.4	13.7	User managed in Grenoble impact of remote backend
replica	local	remote	16.1	16.5	17.0	9.2	9.2	9.4	User managed in Sophia impact of remote backend

Table 6. Experiments for a partial replicated approach between `o=grenoble` and `o=sophia`

Our experimental results for this architecture are presented in tab.6. It can be seen that this approach (with local backends) guarantees the performances with regard to the centralized model (see §4.3). In addition, contrary to the referrals approach, this architecture is able to solve the problem of availability (if a cluster is disconnected, the authentication system is still running). Security is ensured by the protocol LDAP itself whereas delegation is due to the tables' distribution in proposed architecture. This system is therefore a particularly good candidate for a robust authentication system in a distributed environment such as Grid5000.

5 μgrid and application to medical data processing

Grid5000 is an experimental platform for grid computing research that is not making any assumption on the middleware to be used. Instead, Grid5000 users are deploying the middleware they need for their research and experiments. We have deployed the μgrid middleware [8] over the Grid5000 infrastructure. μgrid is a lightweight middleware prototype that was developed for research purposes and already used to deploy applications to medical image processing [9].

The μgrid middleware was designed to use clusters of PCs available in laboratories or hospitals. It is intended to remain easy to install, use and maintain. Therefore, it does not make any assumption on the network and the operating system except that independent hosts with private CPU, memory, and disk resources are connected through an IP network and can exchange messages via communication ports. This matches the Grid5000 platform. The middleware provides the basic functionalities needed for batch-oriented applications: it enables transparent access to data for jobs executed from a user interface. The code of μgrid is licensed under the GPL and is freely available from the authors web page.

The application considered in our experiments is an application to medical image database analysis that is further detailed in [9]. The objective is too face the huge amount of medical data that one can store on a grid by providing a medical image search tool. Medical images are stored together with accompanying metadata (information on patients, acquisition devices, hospitals, etc). The structure of medical data and metadata is often complex and there are very strong privacy constraints applying on both of them: only a very limited number of authorized people should be able to access medical data content. This is even more critical on a grid infrastructure given the data dispersion and the number of users with a potential access to the data.

To retrieve a medical data, a selection is first done on the associated metadata. This enables queries such as "find the MR Image of Mr Patient, acquired yesterday in this hospital". However, there are many clinical cases where a physician would like to be able to find relevant and similar medical cases to an image he is studying to confirm his/her diagnosis. Given the tremendous amount of medical images produced daily, it is impossible for the user to manually browse through the whole medical image database. An hybrid request is need insteadto perform that kind of query: first some potential candidate images are selected using a query on metadata, and then the candidates are compared to the sample image through a compute intensive image analysis step. A grid is well adapted to handle the computation involved as the images may be distributed over the grid nodes for parallel computations. The kind of image analysis to apply is very dependent on the clinical domain and the features of interests. In [9], we used simple similarity measurements algorithms that give relevant results when looking for visually similar images. Thanks to the gridification of this application, very significant speed-up can be achieved in using a grid infrastructure (it highly depends on the amount of resources available). Thanks to a strict authentication procedure, it is possible to identify users and to check whether they are authorized to access the data to ensure medical privacy.

6 Conclusion

The experiments done confirmed the advantages of the NIS system as a fast deployed and efficient solution. It does perfectly fit the requirements of a cluster where security constraints are weak. Yet, the grid context and more precisely the Grid5000 platform places those constraints in the foreground. The geographic distance which separates the sites does not allow to use a dedicated and controlled network. It is then necessary to ensure the confidentiality of the exchanged information at the level

of the used protocols. LDAP, and more exactly the `ldaps` protocol, supplies this feature. Passing from clusters to grids opens also organization issues between distinct administrative entities. In that case, a centralized model (either based on NIS or LDAP) no longer applies: administrators of each site want to manage their own users. Delegation can be obtained by several ways (see [10, 3] for instance). Here, we split the LDAP namespace across multiple servers and arrange LDAP servers in a hierarchy following the administrative domains (see fig 11). Each site hosts a master server for its relative branch. Then, reaching the information contained in other servers can be done either by the "referrals" mechanism of by replication. Both solutions are possible, but the grid context raises the availability issue too. High-availability is especially critical for enterprise and grids authentication services, because in many cases the system will come to a stop when authentication stops working. That's why a solution based on a partial replication is proposed: in a normal configuration, a referral on a disconnected server jams the authentication of a user handled by this server. This is not the case with a local replication. To sum up, the proposed infrastructure supports heterogeneity, compensates the security flaws of NIS and solves the security, availability and delegation constraints required in the Grid5000 platform which adopted this system.

Evolutions are already planned with the integration of additional information in the LDAP directory such as installed softwares and cluster configurations. The objective is to create a repository used by grid services such as DNS, the monitoring and discovery service or the batch scheduler. We are also looking toward the evaluation of a referral based solution using the OpenLDAP proxy cache[11]. A comparison to Globus and more precisely GSI is also planned.

Finally, the authors want to thank Olivier Richard, Nicolas Capit and Julien Leduc from the ID-IMAG Laboratory for their technical contribution.

References

1. Foster, I., Kesselman, C.: Globus: A metacomputing infrastructure toolkit. International J. of Supercomputer Applications and High Performance Computing **11** (1997) 115–128
2. Fedak, G., Germain, C., N'eri, V., Cappello, F.: Xtremweb: A generic global computing system. In: In IEEE Int. Symp. on Cluster Computing and the Grid. (2001)
3. Foster, I.: The anatomy of the Grid: Enabling scalable virtual organizations. Lecture Notes in Computer Science **2150** (2001)
4. Foster, I., Kesselman, C., Tsudik, G., Tuecke, S.: A Security Architecture for Computational Grids. In: Fifth ACM Conference on Computer and Communications Security Conference, San Francisco, California (1998) 83–92
5. Nowell Corporation: Netware 6 (2005) `http://www.novell.com/documentation/nw6p/index.html`.
6. Wahl, M., Howes, T., Kille, S.: RFC 2251 - Lightweight Directory Access Protocol (v3). Technical report, IETF (1997) `http://www.ietf.org/rfc/rfc2251.txt`
7. Stonebraker, M., Aoki, P.M., Devine, R., Litwin, W., Olson, M.A.: Mariposa: A new architecture for distributed data. In: International Conference on Data Engineering (ICDE). (1994) 54–65
8. Seitz, L., Montagnat, J., Pierson, J.M., Oriol, D., Lingrand, D.: Authentication and autorisation prototype on the microgrid for medical data management. In: Healthgrid'05, Oxford, UK (2005)
9. Montagnat, J., Breton, V., Magnin, I.: Partitionning medical image databases for content-based queries on a grid. Methods of Information in Medicine **44** (2005)
10. Varrette, S., Roch, J.L., Denneulin, Y., Leprevost, F.: Secure Architecture for Clusters and Grids. In IEEE, ed.: Proceedings of the 2ème Conférence Internationale sur les Infrastructures Critiques CRIS 2004), Grenoble, France (2004)
11. Apurva, K.: The OpenLDAP Proxy Cache. Technical report, IBM Research lab of India (2003)