Subject: **A Hierarchical GRID Information Service based on MDS 2.1**

Author: **Michel JOUVIN (jouvin@lal.in2p3.fr)**
Irakli MANDJAVIDZE (Irakli.MANDJAVIDZE@cea.fr)
Partner: **IN2P3 LAL, CEA Saclay, France**

Diffusion: **Public release**

1. ABSTRACT

This note describes an implementation of a hierarchically structured GRID Information Service (GIS) using the 2nd Alpha release of the Metacomputing Directory Service 2.1 from the Globus toolkit. Organization of the information infrastructure, based on the Testbed0 deployment of the DataGRID project, is presented. Configuration of information providers at different levels of the hierarchy is given. Some test examples are provided to check correct functionality of GIS. The viability of the MDS 2.1 based GIS in the Testbed1 phase of the DataGRID project is shown.

2. BACKGROUND

Within the Work Package 6 (WP6) of the DataGRID project the Globus[1] toolkit has been used to implement a multi-national, multi-site testbed of a computational grid, called Testbed0. The GRID Information Service (GIS) deployed within the testbed has hierarchical infrastructure that is schematically shown on Figure 1. The lowest level in this hierarchy is formed from Grid Resource Information Services – GRISs. They collect and report information about available resources on a computational grid, e.g. type of the resources (single processing node, batch cluster…), their processing power, type and version of their operating systems, etc. The next upper levels of the hierarchy are formed from the Grid Index Information Services (GIIS). They provide access points to explore the underlying computational grid. A GIIS at any level consolidates information about resources available underneath as reported by GRIS or GIIS lower in the hierarchy. For example, a site GIIS provides information about all GRID resources available on the site. A regional or country level GIIS delivers information about all the resources that are contributed by the region or the country. Finally, the top level GIIS gives access to the information about the whole computational GRID.

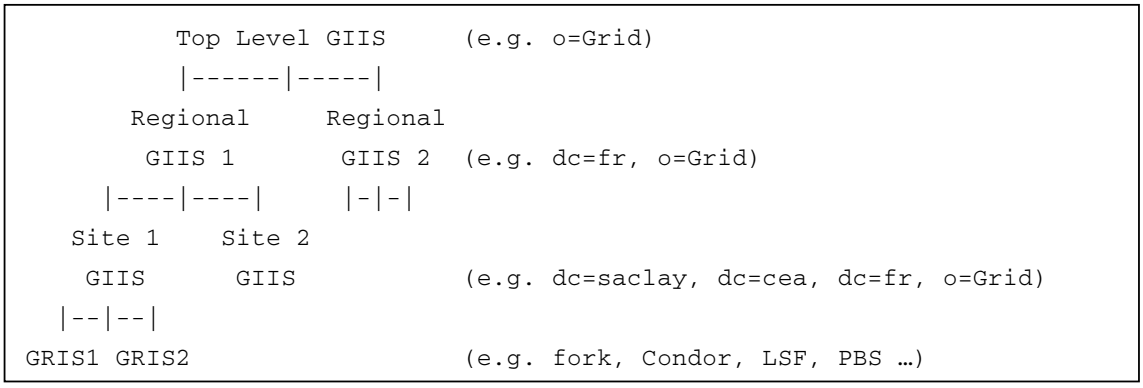

Figure 1 Hierarchical Infrastructure of the GRID Information Service in Testbed0

[1] http://www.globus.com

The Globus toolkit component implementing the GRID information service is called Metacomputing Directory Service (MDS) [1]. The MDS uses the Lightweight Directory Access Protocol (LDAP) [2] and publicly available OpenLDAP[2] software to store and to publish information about resources of a computational GRID and to allow search and query of the information.

By default, the Globus toolkit supports a two-level GIS formed from several GRISs and a GIIS, called the site GIIS. The MDS requires that the GRISs periodically register themselves at the site GIIS. The only information that is transferred during the registration indicates how to perform a query operation on the underlying GRISs (e.g. host name and port). A GRID user can send a query either directly to a specific GRIS or to the site GIIS. In this latter case, the site GIIS issues a search command to all registered GRISs. The GRISs prepare complete information about computing resources that they serve and send it to the site GIIS. The latter delivers all collected information to the user.

For performance reasons, to achieve fast response times and low network load, the GRISs and the site GIIS cache the information describing GRID resources. Depending on its type the lifetime of the cached information varies. If the cache is valid at the time of a search request, its content is returned.

To implement the hierarchical GIS infrastructure shown on Figure 1 for the multi-national and multi-site Testbed0, the default configuration of MDS in the Globus toolkit has been extended basically following the guidelines described in [3]. The new scheme implies that as GRISs, the GIISs at any level of the hierarchy (except the top one) periodically register themselves at a next higher-level GIIS. A GRID user can send a query to a GIIS at any level. The GIIS issue a search command to all registered information providers that on their turn propagate the search operation to underlying layers until GRISs. For performance reasons, the GIISs implement the caching mechanism described above. Obviously, the amount of cached information is growing towards higher levels of hierarchy.

Testbed0 is based on the release 1.1.3 of the Globus toolkit that comes with the version 2.0 of MDS. Recently, the Globus team has made available 2nd Alpha release of MDS 2.1 as a separate package that can be installed and deployed "independently" from the rest of the Globus toolkit. Some security features have been introduced in MDS 2.1. In addition, MDS 2.1 holds a promise to have better performance, to be more reliable and extensible. As activities within the WP6 evolve to a new Testbed1 phase, there is an opportunity to move to MDS 2.1. According to our preliminary testing, deployment of the MDS 2.1 based GRID information service in Testbed1 is a viable option. In this study we investigate the security aspects of MDS 2.1 and how to implement with it a hierarchically structured information service.

3. SECURITY IN THE CURRENT 2ND ALPHA RELEASE OF MDS 2.1

The GRID Information Service implemented with the Globus 1.1.3 toolkit does not require from users an authorization to consult and to acquire information about the GRID. This is mainly because the MDS in the toolkit is based on the OpenLDAP software supporting only version 2 of the LDAP protocol, which lacks mechanisms for strong authentication of LDAP clients and servers.

MDS 2.1 uses a new release of the OpenLDAP software that implements version 3 of the LDAP protocol. The Simple Authentication and Security Layer (SASL) has been added to the latter to overcome the authentication shortcomings of the previous version. SASL in MDS 2.1 is based on the GRID Security Infrastructure (GSI) from the Globus toolkit, which provides secure authentication and communication through the use of X.509 certificates and the Secure Sockets Layer (SSL) protocol.

The security features introduced in MDS 2.1 aim at protection of Grid Information Service from accesses issued by unauthorised users. The baseline security architecture assumes that an information provider (GIIS or GRIS) may specify credentials that must be presented to access the information, which it maintains. In addition, the authenticity of registration messages exchanged among the

[2] http://www.OpenLDAP.org

information providers in a hierarchical GIS has to be ensured and the authorization to perform a given type of registration has to be checked.

In the current 2nd Alpha release of MDS 2.1 some of these security enhancements have been introduced at the GIIS level, but not yet at the GRIS level. A GIIS can be configured to serve search requests only from authorized users. In this case, a user has to provide his/her X.509 certificate to the GIIS (this is done transparently by `grid-info-search` command if the user has a valid GRID proxy). The GIIS validates the user certificate and checks if the user is authorised to query GIS by mapping the user's authentication ID to an entry in the "`gridmap`" file. Note that in order to proceed with the authentication the GIIS itself needs to get a valid X.509 certificate. For compatibility reasons, it is possible to run GIIS in the former non-secure mode.

As mentioned in the previous section, to construct a hierarchical GIS every information provider at a given level has to periodically register itself to a GIIS at its next upper level (see Figure 1): this is called MDS Registration. In the current implementation of the MDS 2.1 no authentication and very little authorization is required for this operation. Therefore it is possible for a malicious user to configure a fake GIIS or GRIS and subscribe to an existing information infrastructure.

It should be reminded that in the current release of MDS 2.1 the information provided by GRIS about underlying computational resources and GRAM (Grid Resource Allocation Manager) depends mainly on configuration files rather than dynamic inspection of the actual configuration. Therefore, accuracy of information registered in configuration file is critical and, again, a malicious user can pollute an existing information infrastructure with inaccurate resource information.

Because the current 2nd Alpha release of MDS 2.1 does not enhance security at the GRIS level, the mechanism used by GIIS to query GRISs has not been modified yet either: a GIIS can only generate anonymous search requests that do not require either authentication or authorization. For the hierarchically structured multi-level GIS this has the following consequences: the Grid Security Infrastructure may protect only the top-level GIIS, while all intermediate level GIISs should operate in the non-secure mode. Any user can generate anonymous search requests that will be served by GRISs and the intermediate level GIISs.

Future (Alpha) releases of MDS 2.1 foresee to extend the GIS protection mechanisms to GRIS. This will allow overcoming current security limitations in the hierarchical GRIS Information Service.

As in the previous version of MDS, the 2nd Alpha release lacks gradation in the information confidentiality. There is no restriction on access to some parts of information based on the user's identity, affiliation or type of requested information. Instead the information is delivered on an all-or-nothing authorization policy (when the authorization procedure succeeds the user receives totality of the information that a GIIS or a GRIS could obtain, even if the user asked only for its subset). Support for fine-grained accesses to GIS information is expected in future releases of MDS.

Finally, current release of MDS 2.1 does not support encrypted communications between the GIS nodes and GRID users.

4. INSTALLATION

The MDS 2.1 package can be obtained from http://www.globus.org/mds2-alpha. Currently available distribution is MDS 2.1 Alpha 2nd release. A single compressed tar-ed file, called "tarball" contains all necessary executable, script and configuration files in a predefined directory tree structure. A user can download the "tarball" file and proceed with installation following the guidelines on the web page above. Another possibility (that has not been tested in this study) is to download the source code distribution that can be obtained from the Globus team on a case-by-case basis. For this an interested user has to be authorised for accesses to the Globus CVS distribution that is protected by GSI. The user should obtain an X.509 certificate from the Globus team and install their CVS client software.

During the tests done at CEA Saclay and IN2P3 LAL the MDS 2.1 package has been installed and configured on PC-s running under RedHat 6.1 and 6.2 Linux distributions. On all these PC-s release 1.1.3 of the Globus toolkit was previously deployed and the `GLOBUS_INSTALL_PATH` environment variable defined. To simplify the discussion Figure 2 shows a sample hierarchical information infrastructure deployed at CEA Saclay (Note that the top-level GIIS running on the seipcd32.saclay.cea.fr host emulates the French country level GIIS).

```
French Country GIIS: host=seipcd32.saclay.cea.fr, port=8890 dc=fr, o=Grid
 Saclay Site GIIS: host=seipca107.saclay.cea.fr,  port=8888,
                                          dc=saclay, dc=cea, dc=fr, o=Grid
  GRIS: host=seipca107.saclay.cea.fr, port=8889 GRAM: fork, condor, pbs
  GRIS: host=seipcd51.saclay.cea.fr,  port=8889 GRAM: fork
  GRIS: host=seipcd52.saclay.cea.fr,  port=8889 GRAM: fork
```

Figure 2 Hierarchical Information Service Infrastructure deployed at CEA Saclay

The configuration at IN2P3 LAL is different (Figure 3). There is only one GRIS that maintains information about a cluster of four Linux PC-s controlled by LSF batch system. The same node runs the GRIS and the LAL site GIIS. The latter registers itself to the existing Testbed0 GIS by joining the IN2P3 wide GIIS that runs on the mardg1.in2p3.fr host at Marseille and is maintained by Testbed0.

```
LAL Site GIIS: host=grid01.lal.in2p3.fr, port=2137,
                                           ou=lal, dc=in2p3, dc=fr, o=Grid
 GRIS: host=grid01.in2p3.fr, port=2135 GRAM: lsf, fork
```

Figure 3 Hierarchical Information Service Infrastructure deployed at IN2P3 LAL

The MDS 2.1 has to be installed under `/opt/globus-mds` directory. This directory should belong to the globus user. First create as root this directory and change its ownership to the globus user, then as the globus user extract contents of the "tarball" file into it. This will create a directory structure as shown on Figure 4.

```
[globus]# ls -l /opt/globus-mds
README bin/ etc/ install/ libexec/ sbin/ share/ tmp/ var/
```

Figure 4 The directory structure of MDS 2.1 extracted from the "tarball" file

5. CONFIGURATION

As can be seen from Figure 2 and Figure 3 a node in a hierarchical GIS can act as a GRIS, a GIIS or both. In any case, the configuration of nodes starts by defining some variables in the `/opt/globus-mds/etc/globus-hostname.conf` and `/opt/globus-mds/etc/grid-info.conf` files. Several configuration files import the environment variables defined in these files and several commands from the Globus toolkit use their values as default parameters.

In the `/opt/globus-mds/etc/globus-hostname.conf` the `GLOBUS_HOSTNAME` variable must be set to the fully qualified hostname of the local node. For example, for the GRIS running on seipcd51.saclay.cea.fr the `GLOBUS_HOSTNAME` variable has the following value:

```
GLOBUS_HOSTNAME ="seipcd51.saclay.cea.fr"
```

In the `/opt/globus-mds/etc/grid-info.conf` file values for the following variables must be defined:

```
SETUP_GRID_INFO_HOST,
SETUP_GRID_INFO_PORT,
SETUP_GRID_INFO_ORGANIZATION_DN.
```

For GRISs and the site GIISs the `SETUP_GRID_INFO_ORGANIZATION_DN` variable should indicate the distinguished name of the site. For an upper level GIIS the variable should refer to the distinguished name of the GIIS domain. Note that distinguished names for sites and GIIS domains should follow the naming convention adopted in WP6[3]. For example, when configuring the GRISs and the site GIIS in CEA Saclay (Figure 2) the following value has been used for the variable:

```
SETUP_GRID_INFO_ORGANIZATION_DN="dc=saclay, dc=cea, dc=fr, o=Grid"
```

For the French site GIIS emulator on seipcd32.saclay.cea.fr this variable has the following value:

```
SETUP_GRID_INFO_ORGANIZATION_DN="dc=fr, o=Grid"
```

In the two-level GIS configuration supported by default in the Globus toolkit the `SETUP_GRID_INFO_HOST` and `SETUP_GRID_INFO_PORT` variables refer to a fully qualified hostname of the node running the site GIIS and its port respectively.

To facilitate configuration of a hierarchical GIS, the following recommendation might be useful.

Recommendation 1 Set the values of the `SETUP_GRID_INFO_HOST` and `SETUP_GRID_INFO_PORT` variables to the host name and port of the GIIS serving the local GIIS domain. (Note, that for a GRIS this will naturally result to the site GIIS hostname and its port values.)

For example, when configuring the GRISs and the site GIIS in CEA Saclay (Figure 2) the following values have been used for these variables on the corresponding nodes:

```
SETUP_GRID_INFO_HOST="seipca107.saclay.cea.fr"
SETUP_GRID_INFO_PORT="8888"
```

For the top level GIIS at CEA Saclay, that runs on the seipcd32.saclay.cea.fr node the variables have been set as follows:

```
SETUP_GRID_INFO_HOST="seipcd32.saclay.cea.fr"
SETUP_GRID_INFO_PORT="8890"
```

5.1. CONFIGURATION OF GRIS

A GRIS may provide information about several Grid Resource Allocation Managers – GRAM-s. A typical example is a node on a GRID that implements UNIX fork type job manager (recommended default by the Globus team) and that provides an interface to a cluster of computational nodes governed by a batch system like Condor, LSF or PBS. For simplicity reasons, first configuration steps will be given for the GRIS that serves a UNIX fork type job manager only. Next the multi-GRAM set-up will be discussed.

5.1.1. Configuration of a GRIS for a UNIX fork GRAM

To configure GRIS that will deliver information about a UNIX fork type job manager the following files have to be modified:

```
/opt/globus-mds/etc/globus-gram-reporter.conf,
/opt/globus-mds/etc/globus-jobmanager.conf.
```

In the `/opt/globus-mds/etc/globus-gram-reporter.conf` file the –dmdn option has to point to the "organization dn" as defined by the `SETUP_GRID_INFO_ORGANIZATION_DN` variable in the `/opt/globus-mds/etc/grid-info.conf` file, but there must not be any white spaces

[3] See, for example, http://marianne.in2p3.fr/datagrid/wp6-fr/giis/giis-france.html#nommage

between the dn components. For example, Figure 5 shows the `/opt/globus-mds/etc/globus-gram-reporter.conf` file that looks the same for all nodes running GRISs at CEA Saclay.

```
[globus /opt/globus-mds/etc]# more globus-gram-reporter.conf
-dmdn "dc=saclay,dc=cea,dc=fr,o=Grid"
-conf /opt/globus-mds/etc/globus-jobmanager.conf
-type fork
-rdn jobmanager
-machine-type unknown
-onetime
```

Figure 5 Example of the globus-gram-reporter.conf file

In the `/opt/globus-mds/etc/globus-jobmanager.conf` file several changes have to be done. The `-globus-org-dn` option has to point to the "organization dn" as defined by the `SETUP_GRID_INFO_ORGANIZATION_DN` variable in the `/opt/globus-mds/etc/grid-info.conf` file, `-globus-gatekeeper-host` option - to the gatekeeper's fully qualified hostname as in the `/opt/globus-mds/etc/grid-hostname.conf` file and `-globus-host-dn` option - to the distinguished name of the gatekeeper. Figure 6 shows an example of `/opt/globus-mds/etc/globus-jobmanager.conf` for the GRIS on seipcd51.saclay.cea.fr.

```
[globus /opt/globus-mds/etc]# more globus-jobmanager.conf
-home /opt/globus-mds
-e  /opt/globus-mds/libexec
-globus-org-dn 'dc=saclay, dc=cea, dc=fr, o=Grid'
-globus-gatekeeper-host 'seipcd51.saclay.cea.fr'
-globus-gatekeeper-port '2119'
-globus-gatekeeper-subject ''
-globus-host-dn 'hn=seipcd51.saclay.cea.fr, dc=saclay, dc=cea, dc=fr, o=Grid'
-globus-host-cputype i686
-globus-host-manufacturer unknown
-globus-host-osname linux
-globus-host-osversion 2.2.15
```

Figure 6 Example of the globus-jobmanager.conf file

The `/opt/globus-mds/etc/grid-info-resource-register.conf` file contains information necessary for GRIS registration with its next upper layer GIIS (i.e. the site GIIS). Normally, this file does not need to be changed as it makes use of variables defined in the `/opt/globus-mds/etc/grid-info.conf` file. However, if for some reason, **Recommendation 1** has not been followed, make sure that the "`reghn`" and "`regport`" variables are set to the site GIIS hostname and its port respectively. The "`hn`" and "`port`" variables in the file should have values of the GRIS hostname and its port respectively. Figure 7 shows a sample `/opt/globus-mds/etc/grid-info-resource-register.conf` file for the GRIS on the seipcd51.saclay.cea.fr node.

```
[globus /opt/globus-mds/etc]# more grid-info-resource-register.conf
dn: service=MDS Resource, hn=${GLOBUS_HOSTNAME}, service=MDS Registration,
${GRID_INFO_ORGANIZATION_DN}
regtype: mdsreg
reghn: ${GRID_INFO_HOST}
regport: ${GRID_INFO_PORT}
regperiod: 300
type: ldap
hn: ${GLOBUS_HOSTNAME}
port: 8889
rootdn: hn=${GLOBUS_HOSTNAME}, ${GRID_INFO_ORGANIZATION_DN}
ttl: 600
timeout: 60
mode: cachedump
cachettl: 30
```

Figure 7 Example of the grid-info-resource-register.conf file

Note that usually the value for the GRIS port defaults to 2135. If for some reasons some other port number has to be used then the `/opt/globus-mds/sbin/SXXgris` start-up script must be modified as well. At the end of the script, the hard-coded value of 2135 in the "`-h ldap://${GLOBUS_HOSTNAME}:2135"` option for the `grid-info-soft-register` script has to be changed by the value of the "`port`" variable defined in `/opt/globus-mds/etc/grid-info-resource-register.conf` (e.g. according to Figure 7 "`-h ldap://${GLOBUS_HOSTNAME}:8889"`).

After these modifications the GRIS can be started. Refer to section 6 for guidelines how to run the MDS 2.1 components.

5.1.2. Configuration with multiple GRAM-s

As discussed above, multiple Grid Resource Allocation Managers can be installed on a single node and GRIS should provide information about each of them. Consider example of the gatekeeper host seipca107.saclay.cea.fr at CEA Saclay, which implements interfaces to Condor and PBS clusters, as well as to the UNIX fork (see Figure 2).

To configure GRIS with an additional GRAM, first a corresponding `globus-jobmanager.conf` file has to be created in the `/opt/globus-mds/etc` directory. Copy the UNIX fork `globus-jobmanager.conf` file and give it a name that reflects the nature of the GRAM, e.g. `globus-jobmanager-pbs.conf` for the PBS batch system. (Note that a configuration file `globus-jobmanager-condor.conf` for Condor exists in the installed distribution.) Edit the `-globus-gatekeeper-subject` option in the newly created file so that its value corresponds to the subject field of the gatekeeper's certificate. In the case of the Condor job manager make sure that `-condor-arch` and `-condor-os` variables have correct values (e.g. `INTEL` and `LINUX` respectively for Linux/PC nodes). Figure 8 illustrates the `globus-jobmanager-pbs.conf` file for the GRIS on seipca107.saclay.cea.fr node.

```
[globus /opt/globus-mds/etc]# more globus-jobmanager-pbs.conf
-home /opt/globus-mds
-e /opt/globus-mds/libexec
-globus-org-dn 'dc=saclay, dc=cea, dc=fr, o=Grid'
-globus-gatekeeper-host 'seipca107.saclay.cea.fr'
-globus-gatekeeper-port '2119'
-globus-gatekeeper-subject
'/CN=seipca107.saclay.cea.fr/OU=DAPNIA/O=CEA/C=FR'
-globus-host-dn 'hn=seipca107.saclay.cea.fr, dc=saclay, dc=cea, dc=fr, o=Grid'
-globus-host-cputype i686
-globus-host-manufacturer unknown
-globus-host-osname linux
-globus-host-osversion 2.2.12
```

Figure 8 Example of the globus-jobmanager-pbs.conf file

Next a corresponding `globus-gram-reporter.conf` file has to be created in the `/opt/globus-mds/etc` directory. Copy the UNIX fork `globus-gram-reporter.conf` file and give it a name that reflects the nature of the GRAM (e.g. `globus-gram-reporter-condor.conf` or `globus-gram-reporter-pbs.conf` for the Condor or PBS batch systems respectively). Several changes have to be done in the newly created GRAM reporter file. The option `–conf` should point to the corresponding job manager configuration file (e.g. `globus-jobmanager-condor.conf` or `globus-jobmanager-pbs.conf`). The `–type` option should reflect the job manager's type (e.g. `condor` or `pbs`). The –rdn option should give the job manager's name (e.g. `jobmanager-condor` or `jobmanager-pbs`). Figure 9 illustrates the `globus-gram-reporter-pbs.conf` file for the GRIS on the seipca107.saclay.cea.fr node.

```
[globus /opt/globus-mds/etc]# more globus-gram-reporter-pbs.conf
-dmdn "dc=saclay,dc=cea,dc=fr,o=Grid"
-conf /opt/globus-mds/etc/globus-jobmanager-pbs.conf
-type pbs
-rdn jobmanager-pbs
-machine-type unknown
-onetime
```

Figure 9 Example of the globus-gram-reporter-pbs.conf file

Next, the `/opt/globus-mds/etc/grid-info-resource-ldif.conf` file has to be modified. Copy the following sequence of lines at the end of the file:

```
dn: hn=${GLOBUS_HOSTNAME}, ${GRID_INFO_ORGANIZATION_DN}
objectclass: GlobusTop
objectclass: GlobusActiveObject
objectclass: GlobusActiveSearch
type: exec
path: /opt/globus-mds/libexec
base: globus-gram-reporter
args: -f /opt/globus-mds/etc/globus-gram-reporter.conf –onetime
cachetime: 30
timelimit: 10
sizelimit: 20
```

And change the name of a configuration file in the `args` variable to the corresponding gram reporter file (e.g. `globus-gram-reporter-condor.conf` or `globus-gram-reporter-pbs.conf` for Condor or PBS batch systems respectively).

Finally, make sure that all paths to the local programs relevant to the corresponding GRAM are correctly set in the `/opt/globus-mds/etc/{$ARCH}/globus-sh-commands.sh` file, where the `ARCH` variable points to the node's architecture (e.g. `i686-pc-linux-gnu`). For illustration purposes, Figure 10 shows the modifications made to the original `globus-sh-commands.sh` file on seipca107.saclay.cea.fr to make it conform to the local PBS system.

```
[globus /opt/globus-mds/etc/i686-pc-linux-gnu]# diff globus-sh-commands.sh
globus-sh-commands.sh.ORIG
<GLOBUS_SH_QDEL="/home/grid/applications/OpenPBS/bin/qdel"
<GLOBUS_SH_QSTAT="/home/grid/applications/OpenPBS/bin/qstat"
<GLOBUS_SH_QSUB="/home/grid/applications/OpenPBS/bin/qsub"
----
>GLOBUS_SH_QDEL=""
>GLOBUS_SH_QSTAT=""
>GLOBUS_SH_QSUB=""
```

Figure 10 Example of the modifications in the globus-sh-commands.sh file for PBS system

5.2. CONFIGURATION OF GIIS

In a multi-level hierarchical GIS, a GIIS at any level receives MDS Registration requests from underlying GRISs or GIISs. On the other hand, a GIIS, except of the top level GIIS, has to periodically register itself with its next upper level GIIS. In addition, with the new security features in MDS 2.1 accesses to a GIIS can be restricted only to authorised users. In the following sub-sections first these aspects will be covered separately, followed by configuration guidelines specific to intermediate and top level GIIS respectively and taking into account actual capabilities of the current 2nd Alpha release of MDS 2.1.

5.2.1. Acceptance of MDS Registration Requests from underlying GIISs

The last line in the `/opt/globus-mds/etc/grid-info-site.conf` file determines which registration should be accepted by a GIIS. Usually this line looks as follows:

```
dn: service=MDS Resource, hn=*, service=MDS Registration,
${GRID_INFO_ORGANIZATION_DN}
```

It stipulates that a GIIS with its domain determined by the `GRID_INFO_ORGANIZATION_DN` variable (in the `/opt/globus-mds/etc/grid-info.conf` file) will accept all MDS Registrations from GRISs or GIISs that belong to the same GIIS domain. This configuration is generally valid for a site GIIS where you want to restrict registration to the GRISs of the same organization (same GIIS domain). However, for upper level GIISs, which federate different branches of a GIS tree, the configuration is no more valid as GIIS domains (determined by the corresponding `GRID_INFO_ORGANIZATION_DN` variables) vary from level to level. To ensure correct operation, the '`dn`' entry in the `/opt/globus-mds/etc/grid-info-site.conf` file of an upper level GIIS should match all possible '`dn`' patterns of MDS registration requests from underlying GIISs.

For example, for the IN2P3 wide GIIS (`${GRID_INFO_ORGANIZATION_DN}="dc=in2p3, dc=fr, o=Grid"`) to accept MDS registration from all site GIISs of IN2P3 laboratories (distinguished by an OU component under the IN2P3 GIIS domain, the '`dn`' entry in its `/opt/globus-mds/etc/grid-info-site.conf` file could be defined as:

```
dn: service=MDS Resource, hn=*, service=MDS Registration, ou=*,
${GRID_INFO_ORGANIZATION_DN}
```

Similarly, the 'dn' entry for the French country level GIIS emulator (domain "dc=fr, o=Grid") running on the seipcd32.saclay.cea.fr node (see Figure 2) has been modified as follows:

```
dn: service=MDS Resource, hn=*, service=MDS Registration, dc=*,
${GRID_INFO_ORGANIZATION_DN}
```

5.2.2. MDS Registration with an upper Level GIIS

An intermediate level GIIS should periodically register its MDS service with its next upper level GIIS. For this, first copy the `/opt/globus-mds/etc/grid-info-resource-register.conf` file to `/opt/globus-mds/etc/grid-info-giis-register.conf` file. This file must contain information necessary for GIIS service registration with its next upper level GIIS. Make sure that the "`reghn`" and "`regport`" variables have values of the upper level GIIS hostname and its port respectively. The "`hn`" and "`port`" variables in the file should have values of the local GIIS hostname and its port respectively, as defined by the `SETUP_GRID_INFO_HOST` and `SETUP_GRID_INFO_PORT` variables in the `/opt/globus-mds/etc/grid.conf` file.

As can be seen on Figure 2 the CEA Saclay site GIIS running on seipca107.saclay.cea.fr (port number 8888) subscribes its MDS service to the top level GIIS running on seipcd32.saclay.cea.fr (port number 8890). A sample `/opt/globus-mds/etc/grid-info-giis-register.conf` file for the site GIIS on seipca107.saclay.cea.fr is shown on Figure 11.

```
[globus /opt/globus-mds/etc]# more grid-info-giis-register.conf
dn: service=MDS Resource, hn=${GLOBUS_HOSTNAME}, service=MDS
Registration, ${GRI D_INFO_ORGANIZATION_DN}
regtype: mdsreg
reghn: seipcd32.saclay.cea.fr
regport: 8890
regperiod: 300
type: ldap
hn: ${GLOBUS_HOSTNAME}
port: 8888
rootdn: hn=${GLOBUS_HOSTNAME}, ${GRID_INFO_ORGANIZATION_DN}
ttl: 600
timeout: 60
mode: cachedump
cachettl: 30
```

Figure 11 Example of the grid-info-giis-register.conf file

Next, the `/opt/globus-mds/sbin/SXXgiis` start-up script has to be modified. Usually, this script launches only the `slapd` stand-alone LDAP daemon that interacts with the underlying GRIS services and serves user's search requests. To start both the `slapd` daemon and the MDS Registration service the `/opt/globus-mds/sbin/grid-info-soft-register` script can be used. Modify the `/opt/globus-mds/sbin/SXXgiis` file changing the line:

```
${libexecdir}/grid-info-slapd -h
ldap://${GRID_INFO_HOST}:${GRID_INFO_PORT} -d 0 -f
${sysconfdir}/grid-info-site-slapd.conf &
```

by the following line:

```
${sbindir}/grid-info-soft-register -log ${localstatedir}/grid-info
system.log -f ${sysconfdir}/grid-info-giis-register.conf --
```

```
${libexecdir}/grid-info-slapd -h ldap://${GLOBUS_HOSTNAME}:8888 -d 0
-f ${sysconfdir}/grid-info-site-slapd.conf &
```

5.2.3. Configuration of GIIS security

The value of `anonymousbind` parameter in the `/opt/globus-mds/etc/grid-info-site-slapd.conf` file determines whether a GIIS will be protected by GSI or not. This variable can take values of "`yes`" or "`no`". To activate the GSI security mechanism for a GIIS, make sure that the `/opt/globus-mds/etc/grid-info-site-slapd.conf` file contains the following line:

```
anonymousbind no
```

Edit the `/opt/globus-mds/libexec/grid-info-slapd` file so that the GRIDMAP variable points to the actual "`gridmap`" file on your system. Figure 12 illustrates `grid-info-slapd` file for the secure top-level GIIS running on the `seipcd32.saclay.cea.fr` node.

```
[globus /opt/globus-mds/etc]# more /opt/globus-mds/libexec/grid-info-slapd
#!/bin/sh

X509_USER_CERT=/opt/globus-mds/etc/LDAP/server.cert
X509_USER_KEY=/opt/globus-mds/etc/LDAP/server.key
LD_LIBRARY_PATH="/opt/globus-mds/install/SASL/lib:/opt/globus-
mds/install/gsi/development/i686-pc-linux-
gnu_nothreads_standard_debug/lib:/opt/globus-
mds/install/openldap/lib:/opt/globus-
mds/install/openssl/lib:${LD_LIBRARY_PATH}"

GRIDMAP=/home/grid/applications/globus/globus-1.1.3/etc/grid-mapfile

export LD_LIBRARY_PATH
export X509_USER_CERT
export X509_USER_KEY
export GRIDMAP
#. /opt/globus-mds/etc/grid-info.conf
exec /opt/globus-mds/install/openldap/libexec/slapd "$@"
```

Figure 12 Example of the grid-info-slapd file for a secure GIIS at CEA Saclay

Only those users who will have an entry in the "`gridmap`" file pointed by the `GRIDMAP` variable in the `/opt/globus-mds/libexec/grid-info-slapd` file will be able to query the GIIS.

To run GIIS in the secure mode, a certificate is required for the GIIS server. The common name in the subject of the certificate must be "`ldap/${GLOBUS_HOSTNAME}`" where the value of the `GLOBUS_HOSTNAME` variable is determined from the `/opt/globus-mds/etc/globus-hostname.conf` file (fully qualified hostname of the GIIS node). For example, the subject of the top level GIIS server running on the seipcd32.saclay.cea.fr node looks as follows:

```
/CN=ldap/seipcd32.saclay.cea.fr/OU=DAPNIA/O=CEA/C=FR
```

The files with the GIIS server's public certificate and private key should be named as `server.cert` and `server.key` respectively and placed under the `/opt/globus-mds/etc/LDAP` directory. The `server.key` file should be readable only for its owner (i.e. the globus user).

Make sure that the `/opt/globus-mds/share/certificates` directory contains public certificates of all Certification Authorities trusted by your site and the `ca-signing-policy.conf` file has corresponding entries and expected conditional subjects. It is possible to link the `/opt/globus-mds/share/certificates` directory to the corresponding directory from the previously deployed and running Globus 1.1.3 toolkit.

5.2.4. Configuration of GIISs on different levels of GIS hierarchy

Configuring GIISs in the hierarchical GRID information infrastructure two cases should be distinguished: the GIIS at the top level of the hierarchy and GIISs at the intermediate levels.

5.2.4.1. A Top Level GIIS

A top level GIIS does not have to run the MDS Registration service as it is on the top of hierarchy. In the current 2nd Alpha release it can be protected to prevent accesses from unauthorized users.

Configuring a top level GIIS:

1) Decide whether it has to be secure or not.
   a. For a secure top-level GIIS follow guidelines in section 5.2.3.
   b. For a non-secure top level GIIS make sure that the `/opt/globus-mds/etc/grid-info-site-slapd.conf` file contains the following line:

      ```
      anonymousbind yes
      ```

2) Following the guidelines in section 5.2.1 make sure that the top level GIIS will accept MDS Registration requests from all GRISs and/or all intermediate level GIISs from the next lower level in the GIS hierarchy.

5.2.4.2. Intermediate Level GIISs

In the current MDS 2.1 release the intermediate level GIIS cannot be protected by GSI and therefore only anonymous binds should be allowed in the `/opt/globus-mds/etc/grid-info-site-slapd.conf` file.

Configuring an intermediate level GIIS:

1) Make sure that the `/opt/globus-mds/etc/grid-info-site-slapd.conf` file contains the following line:

   ```
   anonymousbind yes
   ```

2) Following the guidelines in section 5.2.1 make sure that the intermediate level GIIS will accept MDS Registration requests from all GRISs and/or all intermediate level GIISs from the next lower level in the GIS hierarchy.
3) Following the guidelines in section 5.2.2 make sure that the intermediate level GIIS will register its MDS service to an upper level GIIS in the GIS hierarchy.

6. RUNNING GRIS AND GIIS

Recommendation 2 The Globus team recommends installation, deployment and maintenance of the Globus toolkit as the globus user. This account must also be used to run Globus components.

The scripts `SXXgiis` and `SXXgris` in the `/opt/globus-mds/sbin` directory are used to start/stop GIIS and GRIS servers respectively. To start/stop a GIIS server run the following commands as the globus user:

```
/opt/globus-mds/sbin/SXXgiis start
/opt/globus-mds/sbin/SXXgiis stop
```

To start/stop a GRIS server run the following commands as the globus user.

```
/opt/globus-mds/sbin/SXXgris start
/opt/globus-mds/sbin/SXXgris stop
```

It is also possible to set up MDS 2.1 components to automatically start up at a system boot time.

7. TESTS

Although the MDS 2.1 2nd Alpha release comes as a separate package from the Globus toolkit 1.1.3 (this should change with the upcoming Globus 2.0 release) an installed and deployed Globus 1.1.3 release is necessary. As MDS 2.1 replaces a subset of Globus 1.1.3 commands, both must be in the user's PATH environment variable with MDS 2.1 before Globus 1.1.3.

Recommendation 3 Though the globus user account is used to install, configure, maintain and run MDS 2.1 components, you should not try to test a GIS as the globus user.

7.1. GETTING INFORMATION FROM SEQURE TOP LEVEL GIIS

Prior to issue search commands to a secure top level GIIS, the user has to create a proxy. For this the `grid-proxy-init` command from the MDS 2.1 distribution has to be used. Figure 13 shows an example of the `grid-info-search` command that queries the secure top-level GIIS at CEA Saclay running on the seipcd32.saclay.cea.fr node.

```
seipcd52{mandjavi}: /opt/globus-mds/bin/grid-proxy-init
Your identity: /CN=Irakli
MANDJAVIDZE/Email=Irakli.MANDJAVIDZE@cea.fr/OU=DAPNIA/O=CEA/C=FR
Enter GRID pass phrase for this identity:
Creating proxy ................................ Done
Your proxy is valid until Mon Sep  3 21:35:30 2001
seipcd52{mandjavi}: /opt/globus-mds/bin/grid-info-search -h
seipcd32.saclay.cea.fr -p 8890 "service=*"
SASL/GSSAPI authentication started
SASL SSF: 56
SASL installing layers
version: 2
# filter: service=*
# requesting: ALL
# jobmanager, hn=seipcd51.saclay.cea.fr, dc=saclay, dc=cea, dc=fr, Grid
dn: service=jobmanager, hn=seipcd51.saclay.cea.fr, dc=saclay, dc=cea, dc=fr,
o=Grid
--- (The output of the grid-info-search command has been truncated)
# search result
search: 6
result: 0 Success
# numResponses: 10
# numEntries: 9
```

Figure 13 Example of grid-info-search on a secure top-level GIIS

The first three lines of the `grid-info-search` output indicate the progression of the SASL layer in the user-GIIS server mutual authentication and in the user authorization to use the GIS.

7.2. GETTING INFORMATION FROM NON SEQURE GIIS AND GRIS

To obtain information from non-secure intermediate level GIISs or directly from GRISs the user has to issue the `grid-info-search` command with –x option. The option indicates that no authentication/authorization is necessary. Figure 14 shows an example of the `grid-info-search` command which queries the GRIS running on the seipca107.saclay.cea.fr node at CEA Saclay.

```
seipcd52{mandjavi}: /opt/globus-mds/bin/grid-info-search -h
seipca107.saclay.cea.fr -p 8889 "service=*" -x
version: 2
# filter: service=*
# requesting: ALL
# jobmanager, hn=seipcd51.saclay.cea.fr, dc=saclay, dc=cea, dc=fr, Grid
dn: service=jobmanager, hn=seipcd51.saclay.cea.fr, dc=saclay, dc=cea,
dc=fr, o=Grid
--- (The output of the grid-info-search command has been truncated) ---
# search result
search: 2
result: 0 Success
# numResponses: 4
# numEntries: 3
```

Figure 14 Example of grid-info-search on a non-secure GRIS

7.3. INTEROPERABILITY WITH THE GIS BASED ON MDS 2.0

During this study, the CEA Saclay and IN2P3 LAL GRID information infrastructures (based on the MDS 2.1 Alpha release) have been integrated to the Testbed0 GIS (deployed on the MDS 2.0). For this the site GIIS at CEA Saclay that runs on seipca107.saclay.cea.fr has been registered with the French country level GIIS running on marianne.in2p3.fr (see Figure 15).

```
Top GIIS MDS 2.0: host=testbed001.cern.ch, port=2167,             o=Grid
 French GIIS MDS 2.0: host=marianne.in2p3.fr, port=2137,   dc=fr, o=Grid
  Saclay GIIS MDS 2.1: host=seipca107.saclay.cea.fr, port=8888,
                                    dc=saclay, dc=cea, dc=fr, o=Grid
   GRIS: host=seipca107.saclay.cea.fr, port=8889 GRAM: fork, condor, pbs
   GRIS: host=seipcd51.saclay.cea.fr,  port=8889 GRAM: fork
   GRIS: host=seipcd52.saclay.cea.fr,  port=8889 GRAM: fork
```

Figure 15 Integration of the GIS at CEA Saclay with the GIS of Testbed0

The search operation to the French country level GIIS can be issued using `grid-info-search` command from Globus 1.1.3 release, as well as from the MDS 2.1 release. Note that in this latter case "`-x -P2`" options have to be used in order to avoid authentication procedure with the non-secure GIIS on marianne.in2p3.fr and to use the LDAPv2 protocol respectively (see Figure 16).

```
seipcd52{mandjavi}54: /opt/globus-mds/bin/grid-info-search -h
marianne.in2p3.fr -p 2137 "service=*" -x -P2
version: 2
# filter: service=*
# requesting: ALL
# jobmanager, hn=seipca107.saclay.cea.fr, dc=saclay, dc=cea, dc=fr, Grid
dn: service=jobmanager, hn=seipca107.saclay.cea.fr, dc=saclay, dc=cea,
dc=fr, o=Grid
--- (The output of the grid-info-search command has been truncated)
```

Figure 16 Example of grid-info-search on French GIS

Interoperability between MDS 2.1 based GRIS and MDS 2.0 based GIIS has been also shown.

8. KNOWN PROBLEMS OF THE MDS 2.1 2ND ALPHA RELEASE

This section describes some problems that can be encountered when testing a GIS based on the 2nd Alpha release of MDS 2.1. Follow Recommendation 3 to facilitate testing of a GIS set-up.

8.1.1. Authentication Errors in the Secure Mode of Operation

When for some reasons user authentication fails on a secure GIIS, a connection between the user process (LDAP client) and the GIIS (LDAP server) cannot be established (binding fails) and the following error message occurs:

```
Ldap_sasl_interactive_bind_s: Local error
```

Troubleshooting the problem can be difficult as the MDS log files on the secure GIIS node and on the user's node do not contain more explicit error messages that can be used to determine the reason of the failure. Here is a list of main reasons for such an error:

1) The user does not have a valid proxy or is not defined in the `gridmap` file.
2) The globus user has a proxy. Delete it with the `grid-proxy-destroy` command.
3) The user has an old proxy obtained by the `grid-proxy-init` command from the Globus 1.1.3 toolkit. It must be deleted and the `grid-proxy-init` command from MDS 2.1 must be used.
4) GIIS server certificate or private key is invalid (e.g. wrong subject)
5) CA that has delivered the GIIS certificate is not recognized. Check if CA is defined in the `/opt/globus-mds/share/certificates/ca-signing-policy.conf` file and add it if necessary. Check also for possible syntax errors in this file.

To illustrate, consider the case when a top level GIIS runs in the secure mode. If the globus user has a proxy (valid or not) on the node that runs the secure GIIS server, any user on any node on the GRID will fail to authenticate with the GIIS. In the example on Figure 17 a secure GIIS server is started on the `seipcd32.saclay.cea.fr` node. Then the `globus` user creates a proxy. The user `mandjavi` on the `seipcd52.saclay.cea.fr` node issues the `grid-info-search` command that fails with the "`ldap_sasl_interactive_bind_s: Local error`" message.

```
seipcd32{globus}: /opt/globus-mds/sbin/SXXgiis start
Starting up Openldap 2.0 SLAPD server for the GIIS
seipcd32{globus}: /opt/globus-mds/bin/grid-proxy-init
Your identity: /CN=GLOBUS/OU=DAPNIA/O=CEA/C=FR
Enter GRID pass phrase for this identity:
Creating proxy ......................... Done
Your proxy is valid until Mon Sep  3 23:45:37 2001

seipcd52{mandjavi}: /opt/globus-mds/bin/grid-info-search -h
seipcd32.saclay.cea.fr -p 8890 "service=*"
SASL/GSSAPI authentication started
ldap_sasl_interactive_bind_s: Local error
```

Figure 17 Example of an authentication error

8.1.2. The grid-proxy-init Command Unable to Create a New Proxy

The MDS 2.1 2nd Alpha release comes with a new version of the `grid-proxy-init` command. By default the `grid-proxy-init` command creates a proxy in the `/tmp` directory under the

x509up_u"user_uid" name where "user_id" is the user's UNIX login id. If a user has such a proxy and wants to create a new one under a different directory and/or name (using the –out option or the X509_USER_PROXY environment variable), the grid-proxy-init command fails to create the new proxy and, instead, recreates the existing one. See for example Figure 18: the user "globus" tries to create a new proxy under ~globus/newproxy, but the proxy internal variable of the grid-proxy-init command has a value of the old proxy's filename.

```
seipcd32{mandjavi}: ls -l /tmp
-rw-------    1 globus   x509up_"globus_id"
seipcd32{globus}: /opt/globus-mds/bin/grid-proxy-init -debug -out
~globus/newproxy
Files being used:
 cert_file: none
 cert_dir : /home/grid/applications/globus/globus-1.1.3/share/certificates
 proxy    : /tmp/x509up_"globus_id"
 user_cert: /home/grid/applications/globus/.globus/usercert.pem
 user_key : /home/grid/applications/globus/.globus/userkey.pem
Your identity: /CN=GLOBUS/OU=DAPNIA/O=CEA/C=FR
```

Figure 18 Malfunctioning of the grid-proxy-init command

There is no such problem with the grid-proxy-init command from the Globus 1.1.3 toolkit but the proxies created by this command are not compatible with MDS 2.1. (The proxies created by the grid-proxy-init command from MDS 2.1 can be used in Globus 1.1.3 deployment).

This behaviour of the new grid-proxy-init command partially explains why the /opt/globus-mds/sbin/globus-setup-test script fails (see Figure 19).

```
seipcd32{mandjavi}: ls -l /tmp
-rw-------    1 globus   x509up_"globus_id"
seipcd32{globus}: /opt/globus-mds/sbin/globus-setup-test
Checking certificate directory ..........done.
Checking user certificate setup .........done.
Checking user key setup ..................done.
Creating proxy certificate ..............
.Enter GRID pass phrase for this identity:
verify OK
....+++++
.............................+++++
.........................................done.
Checking user proxy setup ...............
The X509_USER_PROXY var is set to
  /opt/globus-mds/tmp/test_proxy.7218.pem
This file does not exist or is unreadable.
```

Figure 19 Malfunctioning of the globus-setup-test script due to the grid-proxy-init command

To fix this problem, remove the globus proxy from the /tmp directory (delete the file from /tmp or use the grid-proxy-destroy command).

8.1.3. The globus-setup-test Script

Even if it is present in the MDS 2.1 distribution, the `globus-setup-test` script cannot be used to test a GIS set-up. The script fails because the MDS 2.1 distribution does not (and should not) include a Globus gatekeeper (see Figure 20).

```
seipcd32{mandjavi}: grid-proxy-destroy
seipcd32{globus}: /opt/globus-mds/sbin/globus-setup-test
Checking certificate directory ..........done.
Checking user certificate setup .........done.
Checking user key setup .................done.
Creating proxy certificate ..............
.Enter GRID pass phrase for this identity:
verify OK
............+++++
.+++++
.........................................done.
Checking user proxy setup ...............done.
/bin/sed: can't read /opt/globus-mds/etc/globus-gatekeepers.conf: No such
file or directory
Testing Completed!
```

Figure 20 Another example of the malfunctioning globus-setup-test script

8.1.4. Running the grid-info-search Command in a Nonwritable Directory

If the user issues the `grid-info-search` command in a directory to which the user does not have the write access permission the following "Permission denied" messages occur as shown in Figure 21 (the user "`mandjavi`" changes its working directory to the user "`globus`" home directory and starts the `grid-info-search` command). This happens because the `grid-info-search` command creates some temporary files in the current working directory.

```
seipcd32{mandjavi}: cd ~globus
/home/grid/applications/globus
seipcd32{mandjavi}: /opt/globus-mds/bin/grid-info-search "service=*"
/opt/globus-mds/sbin/config.guess: dummy-11071.c: Permission denied
/opt/globus-mds/sbin/config.guess: dummy-11071.c: Permission denied
-- (Several lines of "Permission denied" messages are truncated here) --
/opt/globus-mds/sbin/config.guess: dummy-11196.c: Permission denied
SASL/GSSAPI authentication started
SASL SSF: 56
SASL installing layers
version: 2
--- (The output of the grid-info-search command is truncated here) ---
```

Figure 21 Example of the "Permission denied" messages

8.1.5. Common Name in GIIS Certificates

The Common Name in the certificate of a secure GIIS server starts with the leading "`ldap/`" chain of characters followed by the server's fully qualified hostname. This creates a problem when retrieving

the server's common name from its certificate by the `grid-cert-info` command from the Globus 1.1.3 toolkit. Consider the example on Figure 22.

```
seipcd32{mandjavi}: grid-cert-info -f /opt/globus-mds/etc/LDAP/server.cert
-subject
/CN=ldap/seipcd32.saclay.cea.fr/OU=DAPNIA/O=CEA/C=FR
seipcd32{mandjavi}: grid-cert-info -f /opt/globus-mds/etc/LDAP/server.cert
-cn
ldap
```

Figure 22 Example of the wrong Common Name output from the grid-cert-info command

It is clear that for any secure GIIS server the `grid-cert-info` command will always return "`ldap`" value for its Common Name component. This is due to the confusion between the "/" inside a common name and the "/" used as a separator.

9. SUMMARY AND CONCLUSIONS

The 2nd Alpha release of MDS 2.1 of the Globus toolkit was investigated. The necessary configuration steps to implement a hierarchically structured GRID Information Service based on the current MDS 2.1 release have been described. The security aspects of such a GIS were studied and its current limitations were indicated. Namely, in the current release of the MDS 2.1 only a top level GIIS can be protected by the GRID Security Infrastructure from accesses of unauthorized users, while all intermediate level GIISs and GRIS can be queried by any user. A malicious user has a possibility to register into GIS information about non-available computing resources. There is no support for a fine-grained access to GIS information; it is delivered to users with an all-or-nothing policy. Interoperability between the GIS-s based on the previous MDS 2.0 and the new MDS 2.1 releases has been shown. Some unexpected and misleading behaviour of the MDS 2.1 based GIS has been documented.

In general, despite of some lack of maturity, the 2nd Alpha release of MDS 2.1 can be used in the Testbed1 phase of the DataGRID project, though more features are necessary to enforce the security of GIS and to bring fine gradation in information confidentiality. The performance enhancements compared to the previous MDS version have to be studied. The feasibility to group GRID users in virtual organizations depending on their professional interests as well as to dedicate some parts of GRID resources to such organizations have to be investigated.

10. AKWNOLEDGEMENTS

Authors would like to thank the Globus developers' team for their help in access to the MDS 2.1 distribution and for discussions about configuration of GIS.

11. REFERENCIES

[1] K. Czajkowski et al., "Grid Information Services for Distributed Resource Sharing", In Proc. 10th IEEE International Symposium on High-Performance Distributed Computing, HPDC-10, IEEE Press, 2001

[2] H. Johner et al., "Understanding LDAP", International Technical Support Organization, SG24-4986-00, http://www.redbooks.ibm.com, June 1998

[3] G. Lo Biondo, "GIIS configuration for INFN sites", http://www.mi.infn.it/~lobiondo/GIS, November 2000