the globus toolkit®
www.globustoolkit.org

Globus Toolkit 2.2
MDS Technology Brief

Draft 4 – January 30, 2003

Summary

The Globus Toolkit 2.2 includes a set of information service components collectively referred to as the Monitoring and Discovery Service (MDS) 2.2[1]. MDS 2.2 simplifies Grid information services by providing a single standard interface and schema for the many information services that are used within a virtual organization.

For most Grid-based projects, we recommend using the MDS as a projectwide standard for publishing and accessing system and application data. The MDS can aggregate information from multiple systems at a given physical site as well as aggregate information from multiple sites within the project. We advocate using cluster-level monitoring solutions, such as Ganglia, for cluster information, and for these mechanisms to advertise their data to the higher tiers of an MDS deployment. We also encourage the use of a Web interface to the MDS, both at a site and at a project level, as a simple means of enabling project members to view system and application information directly.

Overview of MDS 2.2

MDS 2.2 is designed to provide a standard mechanism for publishing and discovering resource status and configuration information. It provides a uniform, flexible interface to data collected by lower-level information providers. It has a decentralized structure that allows it to scale, and it can handle static or dynamic data. A project can also restrict access to data by combining GSI (Grid Security Infrastructure) credentials and authorization features provided by MDS.

The MDS reduces the number of mechanisms required for accessing system information. Local systems may use a wide variety of information-generating mechanisms, but users need to learn how to use only one—MDS—to gain access to the information. The MDS also can make it easier for system developers to provide new information services. This capability is consistent with the "hourglass" role played by most of the Globus Toolkit's components: MDS is the neck of the hourglass, with applications and higher-level services (such as brokers) above it and local information sources below it, as shown in Figure 1. Both sides need work only with MDS, so the number of interactions, APIs, and protocols that need to be used are greatly reduced.

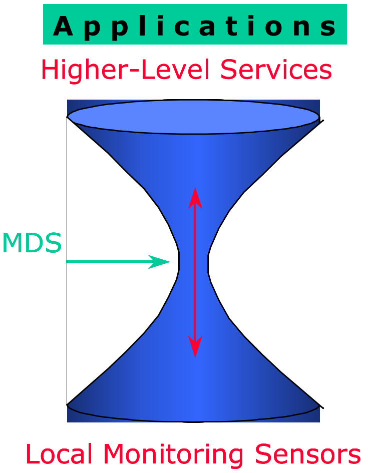


Figure 1: The MDS simplifies interactions between local monitoring mechanisms and higher-level services and applications.

The MDS has a hierarchical structure (see Figure 2) that consists of three main components. The *Grid Index Information Service (GIIS)* provides an aggregate directory of lower-level data. The *Grid Resource Information Service (GRIS)* runs on a resource and acts as a modular content gateway for a resource. *Information Providers (IPs)* interface from any data collection service and then talk to the GRIS. A GRIS registers with a GIIS, and one GIIS may register with another using a soft-state protocol that allows dynamic cleaning of dead resources. Each level also has caching to minimize the transfer of up-to-date data and lessen network overhead.

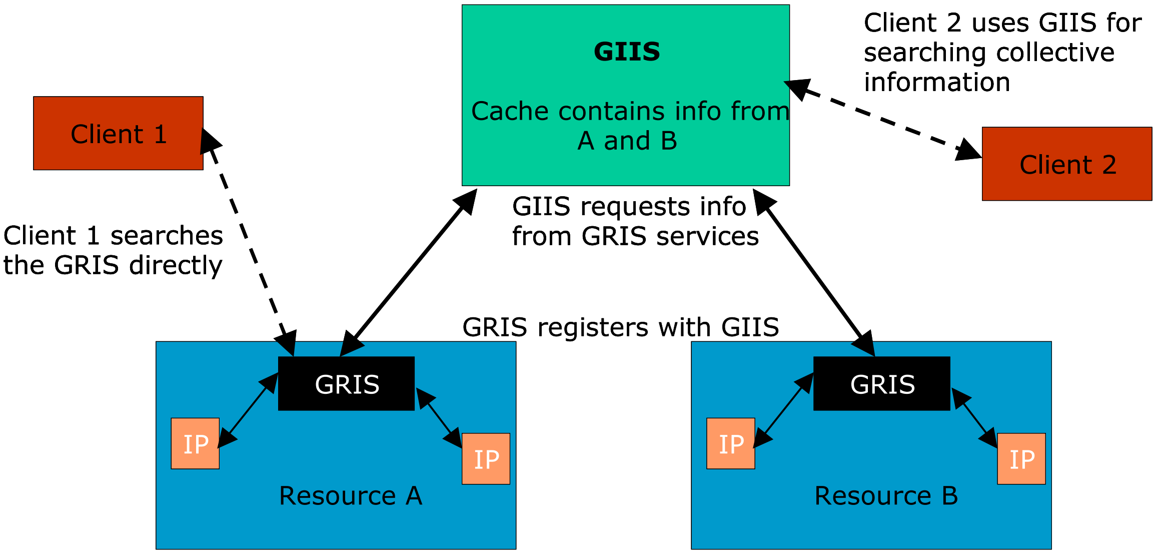
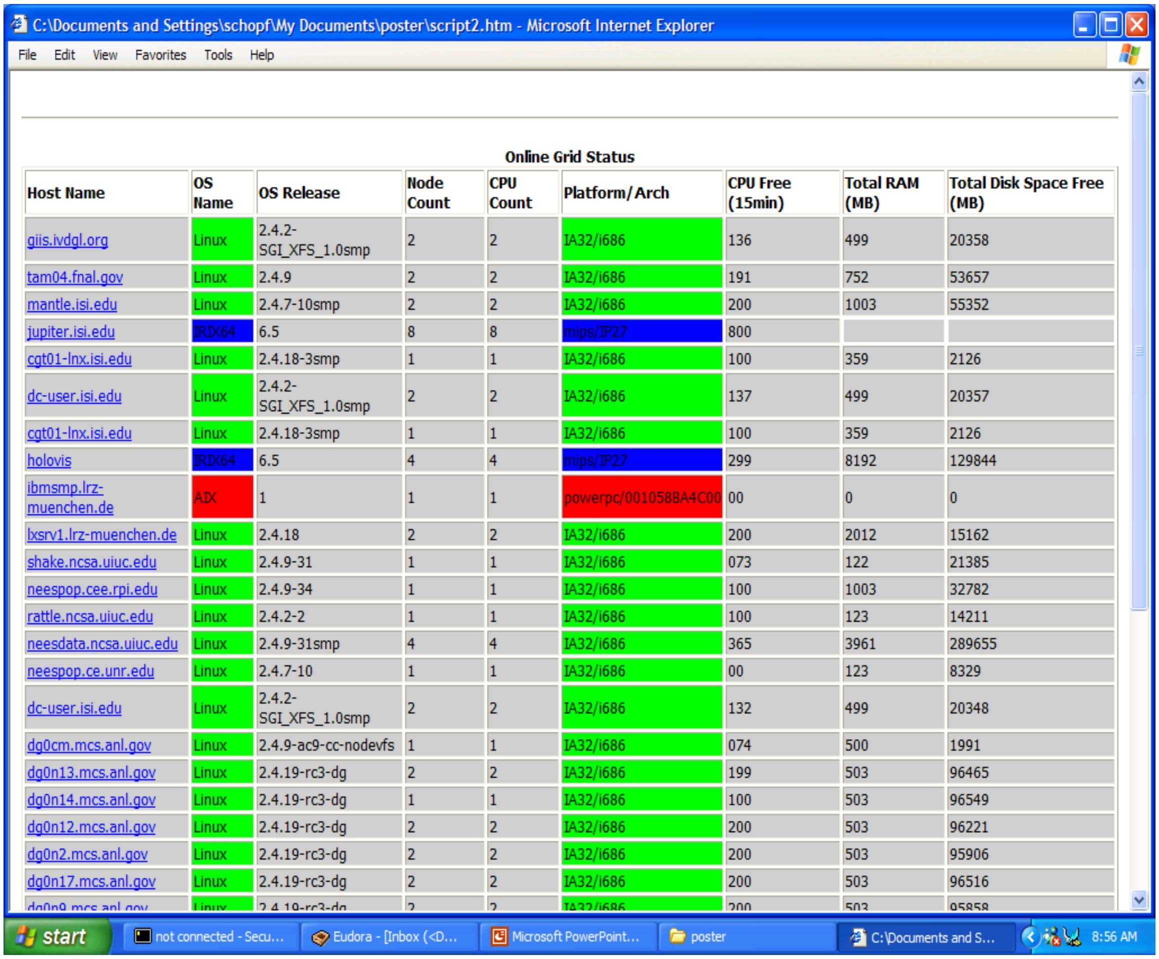

GIIS
Cache contains info from A and B
Client 2 uses GIIS for searching collective information
Client 1
Client 2
Client 1 searches the GRIS directly
GIIS requests info from GRIS services
GRIS registers with GIIS
GRIS
IP
IP
Resource A
GRIS
IP
IP
Resource B

Figure 2: The MDS architecture is a flexible hierarchy. There can be several levels of GIISes, and any GRIS can register with any GIIS, and any GIIS can register with another, making this approach modular and extensible.

Data provided by the core set of MDS information providers includes current load status; CPU configuration; operating system type and version; basic file system information, including free disk space; RAM and virtual memory; and NIC and network interconnect. An additional set of information is available through the GRAM reporter module to advertise job status and queue information. These information providers can provide information both in the MDS core schema and in the GLUE schema, based on a configuration setting. The information published is the same; the difference is in the language it is advertised in, so to speak, the structure and naming is different.

MDS data can be viewed in a number of ways. Any standard LDAP browser can be used to view data from a GIIS or GRIS.[3] User-friendly Web browser access can be provided through a set of PHP scripts on a PHP-enabled Web server.[4] The PHP scripts can be added to any Web page and perform MDS queries to gather basic information. The scripts can be easily adapted to show the summary data needed by a project. All of the information for a given host can also be viewed by clicking on the host name. Figure 3 shows a basic summary provided by these PHP scripts.



Online Grid Status

Host Name	OS Name	OS Release	Node Count	CPU Count	Platform/Arch	CPU Free (15min)	Total RAM (MB)	Total Disk Space Free (MB)
giis.ivdgl.org	Linux	2.4.2-SGI_XFS_1.0smp	2	2	IA32/i686	136	499	20358
tam04.fnal.gov	Linux	2.4.9	2	2	IA32/i686	191	752	53657
mantle.isi.edu	Linux	2.4.7-10smp	2	2	IA32/i686	200	1003	55352
jupiter.isi.edu	IRIX64	6.5	8	8	mips/IP27	800		
cgt01-lnx.isi.edu	Linux	2.4.18-3smp	1	1	IA32/i686	100	359	2126
dc-user.isi.edu	Linux	2.4.2-SGI_XFS_1.0smp	2	2	IA32/i686	137	499	20357
cgt01-lnx.isi.edu	Linux	2.4.18-3smp	1	1	IA32/i686	100	359	2126
holovis	IRIX64	6.5	4	4	mips/IP27	299	8192	129844
ibmsmp.lrz-muenchen.de	AIX	1	1	1	powerpc/001058BA4C00	00	0	0
lxsrv1.lrz-muenchen.de	Linux	2.4.18	2	2	IA32/i686	200	2012	15162
shake.ncsa.uiuc.edu	Linux	2.4.9-31	1	1	IA32/i686	073	122	21385
neespop.cee.rpi.edu	Linux	2.4.9-34	1	1	IA32/i686	100	1003	32782
rattle.ncsa.uiuc.edu	Linux	2.4.2-2	1	1	IA32/i686	100	123	14211
neesdata.ncsa.uiuc.edu	Linux	2.4.9-31smp	4	4	IA32/i686	365	3961	289655
neespop.ce.unr.edu	Linux	2.4.7-10	1	1	IA32/i686	00	123	8329
dc-user.isi.edu	Linux	2.4.2-SGI_XFS_1.0smp	2	2	IA32/i686	132	499	20348
dg0cm.mcs.anl.gov	Linux	2.4.9-ac9-cc-nodevfs	1	1	IA32/i686	074	500	1991
dg0n13.mcs.anl.gov	Linux	2.4.19-rc3-dg	2	2	IA32/i686	199	503	96465
dg0n14.mcs.anl.gov	Linux	2.4.19-rc3-dg	1	1	IA32/i686	100	503	96549
dg0n12.mcs.anl.gov	Linux	2.4.19-rc3-dg	2	2	IA32/i686	200	503	96221
dg0n2.mcs.anl.gov	Linux	2.4.19-rc3-dg	2	2	IA32/i686	200	503	95906
dg0n17.mcs.anl.gov	Linux	2.4.19-rc3-dg	2	2	IA32/i686	200	503	96516
dg0n9.mcs.anl.gov	Linux	2.4.19-rc3-dg	2	2	IA32/i686	200	503	95858

Figure 3: A Web interface to a project's MDS 2.2 GIIS provides a summary of the systems involved in the project and some of their basic status data.

Basic MDS 2.2 Deployment

The basic MDS 2.2 deployment described here is appropriate for most Grid-based projects. It includes the smallest set of features that we believe are needed in order to begin getting a positive “return on investment” from the deployment effort. Naturally, many projects will go well beyond this scenario to provide advanced information services, using additional components from MDS 2.2 or other sources.

Project-Level Configuration

The project should provide at least one project-level GIIS. If redundancy is desired, project-level GIISes may be run at any or all of the project sites. We also recommend installing a Web interface to the MDS on the project’s Web server, configured to display data from the project-level GIIS. This Web interface will provide a simple means for project members to view system and application information directly. The PHP scripts described above can be used to insert MDS data inside existing project web pages.

Site-Level Configuration

If a large number of systems are in use by the project (more than 32 is a reasonable guideline) or if site-specific visibility is important within the project, each site can provide its own site-level GIIS that will register with the project-level GIISes. At the site level we also recommend deploying a Web interface to the MDS.

Resource-Level Configuration

Each resource that provides services to the project (compute, storage, or application services) should provide an MDS GRIS. For example, a cluster should have the GRIS running on the cluster’s head node. Each GRIS should be configured to register with the local site-level GIIS if one exists or, if no site-level GIIS exists, then with all project-level GIISes directly.

We recommend that the GRAM Reporter[5] component of the Globus Toolkit 2.2 be installed and configured with the local GRIS to provide queue information on any systems that use queuing or scheduling systems (PBS, LSF, LoadLeveler, etc.).

For clusters, we recommend installing a cluster-specific monitoring system such as Ganglia.[6] A cluster-specific monitoring system can collect detailed cluster configuration data and communicate it to the GRIS through an MDS information provider. Since an information provider is currently available to interface between Ganglia and the MDS, we encourage the use of this system. If other cluster-monitoring software (such as Nagios or CluMon) is used, however, an information provider can be written with moderate effort.[7]

Capabilities of a Basic MDS 2.2 Deployment

Some capabilities provided by the basic MDS 2.2 deployment are listed below. These capabilities (and many more) are readily available to project members, higher-level software services, and software applications. In addition to the basic Web interface described above, MDS data can be accessed programmatically from a wide range of languages and application development systems.[8] These capabilities are available either form the PHP script Web browser or from the command line:

1. List all of the sites where project-related systems reside.
2. List all of the systems at a given project site and some basic information about them.
3. List all of the systems involved in the project, regardless of where they are located.
4. Given the hostname of a cluster, obtain node count and processor information to decide whether the cluster has enough nodes to handle a parallel application.
5. Given a hostname of a compute node, obtain system memory info to decide whether the node has enough memory to handle a given job.
6. Given a hostname of a storage node, obtain NIC info to decide whether the node can push/pull data fast enough for the application.

Advanced Configurations

Once a basic MDS installation is set up, several advanced configurations are possible. One configuration might include additional information providers, if a cluster-monitoring mechanism other than Ganglia was being used or if the project's applications offered application-specific status information. These information providers would extend the information available via the MDS in project-specific areas.

Another configuration might involve specialized index services (GIIS) for specific needs such as systems management, job scheduling, general resource browsing, or specialized monitoring. The hierarchy described above is best for general resource discovery, but a specialized GIIS could allow additional flexibility.

[1] Details about MDS are available on the Globus Project Website at http://www.globus.org/mds/.

[2] Ganglia is an open source cluster monitoring system available on SourceForge at http://sourceforge.net/projects/ganglia/.

[3] A variety of software tools that provide access to MDS data are listed at http://www.globus.org/mds/getmdsdata/cmdsdata.html.

[4] These PHP scripts are available from http://gldap.mcs.anl.gov/neillm/mdsweb/. Documentation is provided at http://www.mcs.anl.gov/~jms/new.html.

[5] Details on how to enable the GRAM Reporter in Globus Toolkit 2.2 are available at http://www.globus.org/gt2.2/admin/guide-configure.html.

[6] Information about using Ganglia with MDS is available at http://www.globus.org/mds/gangliaprovider.html.

[7] Details on how to develop an MDS information provider are available at http://www.globus.org/mds/creating_new_providers.pdf.

[8] MDS is based on the LDAP protocol, so these capabilities are available in any language or application development system that provides access to LDAP information services.