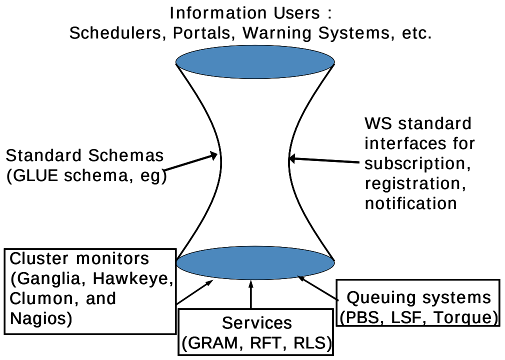
MDS4: THE GT4 MONITORING AND DISCOVERY SYSTEM

Contact email: jms@mcs.anl.gov
Web: http://www.globus.org/toolkit/mds

The Globus Toolkit's Monitoring and Discovery System (MDS4) implements a standard Web services interface to a variety of local monitoring tools and other sources.

MDS4 is a "protocol hourglass," defining standard protocols for information access and delivery and standard schemas for information representation. Below the neck of the hourglass, MDS4 interfaces to different local information sources, translating their diverse schemas into appropriate XML schema (based on standards such as the GLUE schema whenever possible). Above the neck of the hourglass, various tools and applications can be constructed that take advantage of the uniform Web service interfaces to those information sources the MDS4 interacts with.

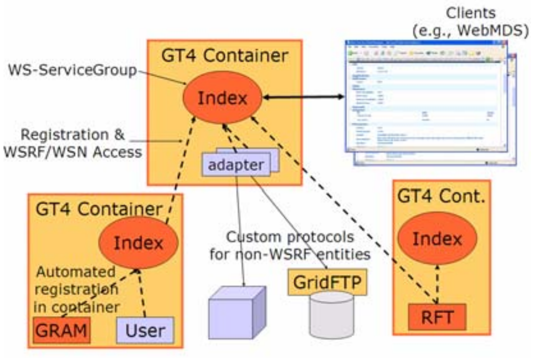


MDS4 builds on query, subscription, and notification protocols and interfaces defined by the WS Resource Framework (WSRF) and WS-Notification families for specifications and implemented by the GT4 Web Services Core.

Building on this base, we have implemented a range of *information providers* used to collect information from specific sources. These components often interface to other tools and systems, such as the Ganglia cluster monitor and the PBS and Condor schedulers.

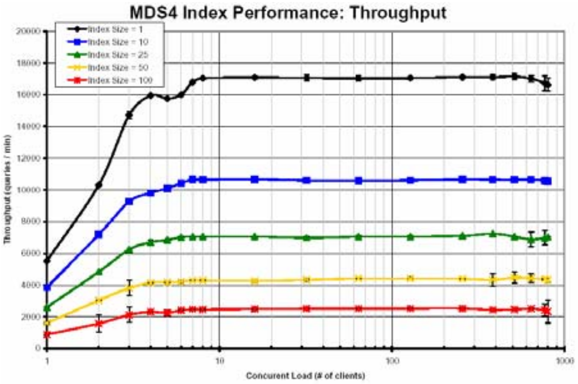
MDS4 also provides two higher-level services: an *Index* service, which collects and publishes aggregated information about information sources, and a *Trigger* service, which collects resource information and performs actions when certain conditions are met. These services are built on the *Aggregation Framework* infrastructure that provides common mechanisms to collect information, be configured, self-clean, and provide a soft consistency.

Additionally, a Web-based user interface called *WebMDS* provides a simple XSLT-transform based visual interface to the data. The figure below depicts a typical MDS4 project deployment.



The GT 4.0.1 Index service has shown a high degree of stability and scalability with respect to the number of concurrent queries. For example, one Index service test ran over 96 days, processing over 623 million queries, averaging 74 per second, with no noticeable performance or usability degradation.

We have also examined the general scalability of the MDS4 Index service with respect to numbers of clients and Index size. We have run LAN experiments on the University of Chicago TeraGrid cluster (dual 2.4 GHz Xeon processors, 3.5 GB RAM, 1 GB/s network) examining index sizes of 1, 10, 25, 50, and 100 and clients, distributed evenly over 20 nodes, from 1 to 800. The figure below shows 8-minute averages of these results.



See also the journal paper available from www.mcs.anl.gov/~jms/Pubs/mds4-scidac.pdf

Finding the Right Resource: Working with the TeraGrid

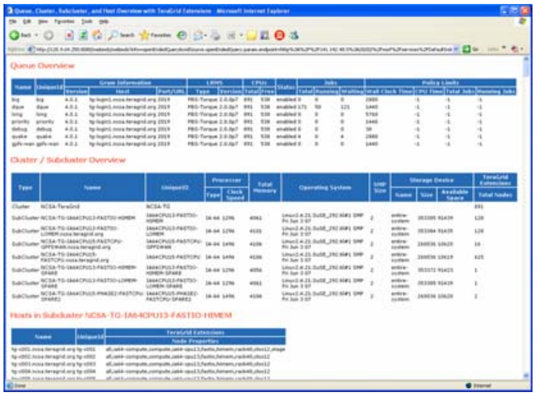
As an example of an MDS4 deployment, we are working with the TeraGrid project to provide data relevant to selecting the appropriate set of resources to use for a particular job. End-users (via a Web interface), metascheduling systems, and other applications will be able to use this deployment to find the resources that best meet their needs.

Information relevant to resource selection is currently available from a variety of sources: cluster monitoring systems, including Ganglia, Clumon, Nagios, and Hawkeye; resource management and scheduling systems, including PBS, Torque, and LSF; and common services, including GRAM, GridFTP, RFT, CAS, and RLS, all part of GT4.

As a result of the current MDS4 deployment, users have access to a simple Web interface (shown right) to examine resource selection decisions. Moreover, metaschedulers have a common interface to the data they need across the TeraGrid, through either command line or Java APIs.

By providing a protocol hourglass, and therefore a level of indirection, we can easily accommodate additional data or changes in the TeraGrid sites simply by adding information providers, with no client-level changes needed.

Current information about the MDS4 deployment on the TeraGrid is available from http://mds.teragrid.org/ .



Queue, Cluster, Subcluster, and Host Overview with TeraGrid Extensions - Microsoft Internet Explorer

Queue Overview

Name	UniqueId	Gram Information			LRMS		CPUs		Status	Jobs			Policy Limits			
		Version	Host	Port/URL	Type	Version	Total	Free		Total	Running	Waiting	Wall Clock Time	CPU Time	Total Jobs	Running Jobs
big	big	4.0.1	tg-login1.ncsa.teragrid.org	2119	PBS-Torque	2.0.0p7	891	538	enabled	0	0	0	2880	-1	-1	-1
dque	dque	4.0.1	tg-login1.ncsa.teragrid.org	2119	PBS-Torque	2.0.0p7	891	538	enabled	171	50	121	1440	-1	-1	-1
long	long	4.0.1	tg-login1.ncsa.teragrid.org	2119	PBS-Torque	2.0.0p7	891	538	enabled	0	0	0	5760	-1	-1	-1
priority	priority	4.0.1	tg-login1.ncsa.teragrid.org	2119	PBS-Torque	2.0.0p7	891	538	enabled	0	0	0	1440	-1	-1	-1
debug	debug	4.0.1	tg-login1.ncsa.teragrid.org	2119	PBS-Torque	2.0.0p7	891	538	enabled	0	0	0	30	-1	-1	-1
quake	quake	4.0.1	tg-login1.ncsa.teragrid.org	2119	PBS-Torque	2.0.0p7	891	538	enabled	4	0	4	2880	-1	-1	-1
gpfs-wan	gpfs-wan	4.0.1	tg-login1.ncsa.teragrid.org	2119	PBS-Torque	2.0.0p7	891	538	enabled	0	0	0	1440	-1	-1	-1

Cluster / Subcluster Overview

Type	Name	UniqueID	Processor		Total Memory	Operating System	SMP Size	Storage Device			TeraGrid Extensions
			Type	Clock Speed				Name	Size	Available Space	Total Nodes
Cluster	NCSA-TeraGrid	NCSA-TG									891
SubCluster	NCSA-TG-IA64CPU13-FASTIO-HIMEM	IA64CPU13-FASTIO-HIMEM	IA-64	1296	4061	Linux2.4.21.SuSE_292.SMP Fri Jun 3 07	2	entire-system	353305	91639	128
SubCluster	NCSA-TG-IA64CPU13-FASTIO-LOMEM.ncsa.teragrid.org	IA64CPU13-FASTIO-LOMEM	IA-64	1296	4101	Linux2.4.21.SuSE_292.SMP Fri Jun 3 07	2	entire-system	353304	91638	128
SubCluster	NCSA-TG-IA64CPU15-FASTCPU-GPFSWAN.ncsa.teragrid.org	IA64CPU15-FASTCPU-GPFSWAN	IA-64	1496	4106	Linux2.4.21.SuSE_292.SMP Fri Jun 3 07	2	entire-system	260036	10629	16
SubCluster	NCSA-TG-IA64CPU15-FASTCPU.ncsa.teragrid.org	IA64CPU15-FASTCPU	IA-64	1496	4106	Linux2.4.21.SuSE_292.SMP Fri Jun 3 07	2	entire-system	260036	10619	615
SubCluster	NCSA-TG-IA64CPU13-FASTIO-HIMEM-SPARE	IA64CPU13-FASTIO-HIMEM-SPARE	IA-64	1296	4056	Linux2.4.21.SuSE_292.SMP Fri Jun 3 07	2	entire-system	353372	91623	1
SubCluster	NCSA-TG-IA64CPU13-FASTIO-LOMEM-SPARE	IA64CPU13-FASTIO-LOMEM-SPARE	IA-64	1296	4061	Linux2.4.21.SuSE_292.SMP Fri Jun 3 07	2	entire-system	353305	91639	1
SubCluster	NCSA-TG-IA64CPU15-PHASE2-FASTCPU-SPARE2	IA64CPU15-PHASE2-FASTCPU-SPARE2	IA-64	1496	4106	Linux2.4.21.SuSE_292.SMP Fri Jun 3 07	2	entire-system	260036	10629	1

Hosts in Subcluster NCSA-TG-IA64CPU13-FASTIO-HIMEM

Name	UniqueId	TeraGrid Extensions
		Node Properties
tg-c001.ncsa.teragrid.org	tg-c001	all,ia64-compute,compute,ia64-cpu13,fastio,himem,rack40,dscl2,stage
tg-c002.ncsa.teragrid.org	tg-c002	all,ia64-compute,compute,ia64-cpu13,fastio,himem,rack40,dscl2
tg-c003.ncsa.teragrid.org	tg-c003	all,ia64-compute,compute,ia64-cpu13,fastio,himem,rack40,dscl2
tg-c004.ncsa.teragrid.org	tg-c004	all,ia64-compute,compute,ia64-cpu13,fastio,himem,rack40,dscl2

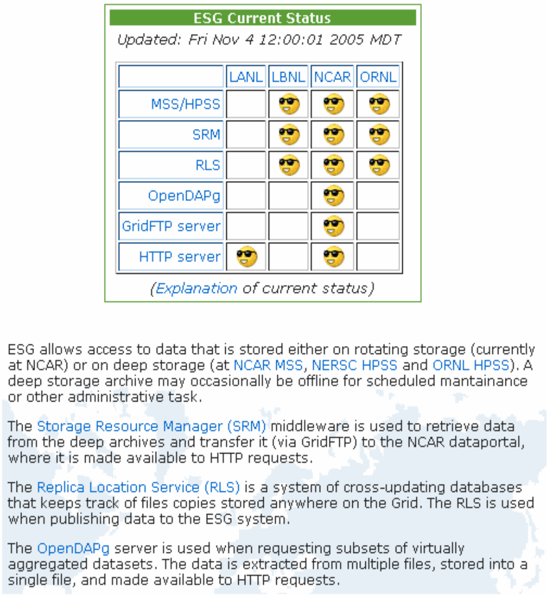
Discovering Errors: Working with the Earth Science Grid

The MDS4 Trigger service collects information and compares that data against a set of conditions defined in a configuration file. When a condition is met, an action takes place, such as emailing a system administrator when the disk space on a server reaches a threshold. This functionality was inspired by a similar capacity in Condor's Hawkeye monitor.

Currently, the Earth Science Grid is using the MDS4 Trigger service to monitor the states of integral service components in that Grid, such as RLS, SRM, OpenDAP, and HTTP and GridFTP fileservers. The Trigger service periodically checks that these services are up and running; if any of these services has gone down or is unavailable for any reason, an action script is executed that sends email to administrators notifying them that the service is unavailable.

The status provided by the Trigger service in ESG is also reflected in the ESG portal Web page, where users can view at a glance the current up/down status of the various component services.

See https://www.earthsystemgrid.org/ to explore the ESG portal page further and view the current system status.



ESG Current Status

Updated: Fri Nov 4 12:00:01 2005 MDT

	LANL	LBNL	NCAR	ORNL
MSS/HPSS		😎	😎	😎
SRM		😎	😎	😎
RLS		😎	😎	😎
OpenDAPg			😎	
GridFTP server			😎	
HTTP server	😎		😎	

(Explanation of current status)

ESG allows access to data that is stored either on rotating storage (currently at NCAR) or on deep storage (at NCAR MSS, NERSC HPSS and ORNL HPSS). A deep storage archive may occasionally be offline for scheduled maintainance or other administrative task.

The Storage Resource Manager (SRM) middleware is used to retrieve data from the deep archives and transfer it (via GridFTP) to the NCAR dataportal, where it is made available to HTTP requests.

The Replica Location Service (RLS) is a system of cross-updating databases that keeps track of files copies stored anywhere on the Grid. The RLS is used when publishing data to the ESG system.

The OpenDAPg server is used when requesting subsets of virtually aggregated datasets. The data is extracted from multiple files, stored into a single file, and made available to HTTP requests.