



# Grid3 Monitoring Meeting and towards OSG-0

Grid3  
2004-xx

August 3-4, 2004

## Table of Contents

1.	Introduction.....	2
2.	Near Term Issues with Existing Systems.....	3
2.1.	Ganglia.....	3
2.1.1.	Program of Work .....	3
2.2.	MonALISA .....	3
2.2.1.	Program of Work .....	4
2.3.	ACDC .....	4
2.3.1.	Program of Work .....	5
2.4.	GridCat.....	6
2.4.1.	Program of Work .....	6
2.5.	MDS Solutions.....	7
2.5.1.	Program of Work .....	7
2.6.	Site Verify.....	8
2.6.1.	Program of Work .....	8
2.7.	VDT Issues.....	9
2.7.1.	MonaLisa version update in VDT.....	9
2.7.2.	Configuration in VDT.....	9
3.	Service Health and Troubleshooting.....	10
3.1.	GridCat.....	10
3.2.	Grid Exerciser .....	10
3.3.	GridScout .....	12
4.	Information Systems .....	12
4.1.	Interoperability with LCG.....	12
4.1.1.	Program of Work .....	13
4.2.	New MDS Systems .....	14
4.2.1.	Program of Work .....	14
4.3.	Use Policy Information.....	14
4.3.1.	Program of Work .....	16
5.	Grid2003 Lessons Coverage.....	16

6.	Observations on OSG-0 .....	17
6.1.	Concepts and Principles .....	17
6.2.	Development/Integration Grid Suggestions for OSG .....	17

## 1. Introduction

These are notes from a meeting of technical experts reviewing the Grid3 monitoring and information systems:

- MDS – the basic information service for Grid3 site attributes.
- MonALISA – the main mechanism for collecting and archiving dynamic information from various Grid3 information providers, including VO statistics for I/O, jobs, and CPU usage.
- ACDC – the job database application from UB that also provides job statistics by VO and keeps an archival history of such statistics.
- MDViewer – the offline application that creates views of metrics collected by the archival service of MonALSA.
- GridCat – the site catalog, display, and diagnostic tool which invokes various scripts to check the status and health of the expected Grid3 services running at a site.
- Ganglia – the out-of-the-box cluster monitoring tool that many sites in Grid3 use to collect usage and performance metrics such as load, cpu consumption, and I/O into and from a site. In the Grid3 context, the issues are mainly support for packaging, configuration, and the top-level collector services.

These are the existing systems. We also discussed where there are gaps in our systems especially in the areas of *troubleshooting*, and *policy information handling*. We heard a presentation on the SAMGrid monitoring system, samTV. Finally, we discussed the OSG project and have made contact with the draft blueprint document.

Since “grid monitoring” is a large, open-ended research area, we identified three primary reasons to monitor in order to frame our discussions: 1) to audit usage and statistics, 2) provide information for scheduling decisions of grid applications; 3) to provide tools for troubleshooting services. The meeting scope was then:

- Review term monitoring issues for Grid3, with an eye towards troubleshooting and gaps in our current systems.
- Review use of information systems MDS and GridCat, possible correlation of information sources, and external monitors of these systems.
- Deliverable to Taskforce and OSG collaboration board a document (this one) describing a program of work for the next 6 months in two parts (a) fixes, upgrades and maintenance of Grid3+ systems; (b) identification of technical direction development in OSG-0.

## 2. Near Term Issues with Existing Systems

### 2.1. Ganglia

It is generally felt by VO users that services and information delivered by Ganglia is useful when running jobs on Grid3, and therefore worthy of continued deployment and support. We identified the following issues:

1. **Packaging** – The current version deployed is 2.5.6, and iVDGL has maintained a Pacman cache for the distribution. At the moment, there is no effort identified to continue support for the Pacman distribution. Note that many Grid3 sites use Ganglia as part of the NPACI Rocks distribution or deploy it directly from the source distribution. The VDT team will reconsider adding the iVDGL provided pacman distribution to the next release.
2. **Support for top level Grid3 and Grid3dev collectors at the iGOC.** Currently the Operations group at IU is maintaining the machine on which the service is deployed. Configuration changes to the displayed list of sites is maintained based upon requests via email notifications which occur during the installation process. The iGOC will continue to support these top level services and provide configuration support for Grid3 site administrators for the purpose of top-level collection.
3. **Configuration, and interface to MonALISA.** The Pacman installation of Ganglia attempts to configuration the most general case configuration. The installation documentation is provided with the main Grid3 install guide. The *gmetad* collector, the *rrdtool* and the web-based interface are deployed on the system, usually the gatekeeper node. Installation of the “*gmond*” sensor is required on each of the compute nodes. Grid3 does not strictly require Ganglia installation for sites. This does cause some problems due to the lack of consistency between the various displays. The interface between MonALISA and Ganglia is scoped entirely within the *configure\_monalisa.sh* installation script which is provided in the VDT release.

#### 2.1.1. Program of Work

In summary, there are no development issues regarding Ganglia systems in Grid3. The program of work consists of packaging and deployment support (the proposal is that VDT absorb this responsibility), continued support for top level services (the iGOC), and continued support for Ganglia-MonALISA interfaces and configuration (the MonALISA team will provide this).

### 2.2. MonALISA

MonALISA provides a global monitoring and information system. It is interfaced with Ganglia to collect compute farm (cluster) monitoring information, and has customized hooks for different local batch queuing systems for jobs (PBS, LSF, Condor) and with data transfers agents such as GridFTP. It is also currently used to perform simple

network measurements among the Grid3 sites. It has been heavily used by VO job submitters for quick views of batch queue levels and status, and its collected information (by the archival services at the iGOC) may be used for more automated site-selectors and grid schedulers. There was expressed a general need for more information regarding its interfaces and internal schema.

### **2.2.1. Program of Work**

The information collected will be extended to :

- Get information from applications and other programs using the ApMon modules (APIs are in c, c++, java, perl, python).
- Network information, topology of the connectivity.
- Alarms and triggers. The system already provides real time and historical access to the monitoring information and it offers the possibility to dynamically deploy filters and alarm triggers. We will deploy soon a set of alarm triggers to assist the grid3 operations team in identifying the components which do not work properly
- We will continue to develop the repository to improve the global views it provides :
  - The status of each site and its components
  - Global statistical page for all the activities
  - Views for utilization of the systems
  - Views for the jobs running at different centers
  - ...

The interactive client allows to access real-time and historical data from all the sites , to access data for each component and to perform aggregation and simple statistical analysis. It also provides a set of predefined global views for the entire system.

The information collected in MonALISA is also available for other services ( using several protocols) able to analyze it (Metrics Data Viewer) or to higher level services which provide decision support.

We are developing a prototype for the STAR experiment Meta-Scheduler, which discovers all the active sites and subscribes to the monitoring information necessary for prediction and decision making mechanisms. The scheduler is a replicated web service system capable to optimize the way jobs are allocated in the system.

Monitors of MonALISA itself should be done within, and from outside MonALISA itself. The ML status page is a view of versions and status of ML components at sites:

<http://monalisa-starlight.cern.ch:8080/stats?page=summary>

### **2.3. ACDC**

The ACDC Job Monitoring system is designed to be an extremely light-weight and non-intrusive tool for monitoring applications and resources on computational grids. It also provides a historical retrospective of the utilization of such resources, which can be used to track efficiency, adjust grid-based scheduling, and perform a predictive assignment of

applications to resources. The intelligent management of consumable resources, including computational cycles, requires accurate up-to-date information. The ACDC Job Monitoring system provides near real-time snapshots of critical computational job metrics, which are stored in a database and utilized by dynamic web pages that it generates for the user. Jobs are segmented into several classes (running, queued, historical, etc.) and statistics for each class are created on-the-fly. Furthermore, all metrics obtained from a given resource is time stamped so that the age of the information displayed or stored is available. In order to present voluminous amounts of data in a hierarchical fashion, all top level ACDC Job Monitoring charts have a "drill down" feature that gives the user increasingly more detailed information about the jobs they are interested in. This feature is essential when post-mortem analysis of historical Grid computational jobs is required for assessing the performance of the Grid (e.g., the number of CPU hours consumed over a given period of time across all available resources).

#### Current Capabilities:

- a. **Running/Queued Jobs:** Summary and statistics of currently running or queued jobs that are on Grid3 resources. Summary charts are compiled for a chart based on {total jobs, total CPU hours or total runtime} for usage data of running or queued jobs for a single or all resources grouped by {user, resource\_vo or queue}. Each chart provides a drill down feature for each bar element to display a table of detailed job information.
- b. **Job History:** Summary and statistics of ~530,000 jobs that have run on Grid3 resources since October 2003. Summary charts are compiled from usage data based on {user jobs, resource\_vo jobs, queue jobs, user nodes, resource\_vo nodes, queue nodes, user runtime, resource\_vo runtime or queue runtime} for a given calendar date range from one day to several years and for a single or all resources. Statistics such as total jobs, average runtime, total CPU time consumed, etc. are dynamically produced from the chart scope defined. Each chart provides a drill down feature for each bar element to display more detailed information. Finally, table of detailed per job information is presented in a table format.
- c. **ACDC Site Status:** Dynamically generated ACDC site status logs report successful monitoring events and also reports specific Grid3 site errors corresponding to monitoring event failures with an appropriate time stamp.

### **2.3.1. Program of Work**

The University at Buffalo is the group responsible development and on-going support of ACDC. Future enhancements to the currently deployed system include:

- a. Provide a programmatic interface to the ACDC Job Monitoring database for running, queued or historical jobs and current site status metrics.
- b. Integrate with MonaLisa and GridCat for defining difference metrics on the job monitoring and other grid metrics.

- c. Intergrate with the MDS provider development team for providing XML formatted job information and site status metrics.
- d. Integrate with grid scheduling team and specifically providing predictive scheduling estimates based on resource policy specifications.
- e. Provide resource specific CPU availability for Grid3 resources and Virtual Organizations.
- f. Provide current availability of free nodes and predictive scheduling capabilities of job execution start times based on running, queued and submitted job characteristics including site policy constraints.
- g. Provide data grid historical and near real-time estimates of bandwidth and utilization of grid-enabled repositories.

## **2.4. GridCat**

The web presentation is designed to be very simplistic way. GridCat installation, configuration, and operation became easier than previous versions. Almost all the requirements that were requested by the taskforce are implemented. For example, GridCat displays available CPU slots and disk information dynamically. Grid test can be performed with standard port or non-standard port. Changes in the site parameters are sometimes recognized during GridCat script cycles. Multiple sites within a facility are displayed with the pi-like dot.

### **2.4.1. Program of Work**

However, there are more features which are required to be implemented in the near term. These features are mainly based on the feedbacks from the collaborators and developers wish lists which were emerged during the development. The new features desirable in the near future include:

- Adding more tables
- Better looking status map
- Disk space should be presented by total or by VO
- Historical data saving or DB snapshot for statistical analysis
- Editable attributes on the web pages based on the grid-proxy
- Better readability for the information on the available resources
- Provide a view for all the information per site

In addition to these, we need to incorporate very low level minor details, e.g., it should also have the ability to provide information on each column in the presentation and what is meant by the specific column. This will provide necessary online help for each item on the catalog page as well as the dictionary for a future reference.

We will need to add these features in less two months of time frame. We will need to analyze what is missing or desirable at that point for future presentation of the GridCat in conferences. One particular and immediate target conference in mind is the CHEP2004 which will be held at the end of September this year.

## 2.5. MDS Solutions

The Grid3 experiences have shown that in performance characteristics of MDS

The important configuration variables for MDS are set in the grid-info-resource-register.conf file they include:

**timeout:** This value determines how long a query to a particular site waits for the requested information to return. During tuning, query times are determined by running timed queries to the site. The current setting for Grid3 is 160s

**cachettl:** This value sets how long the values provided by the information provider remains valid in the cache. A few factors go into determining this value including how static or dynamic the information required from MDS is. The current setting for this parameter in Grid3 is 600s

**Deployment:** MDS information turned off recently in Grid3 for prevention of potential security breach. We have decided in Grid3 to eradicate one level of indexing at the VO level. There will be a top-level GIIS at the IGO and all sites will register to this GIIS, this will essentially reduce human error and neglect at the VO level which always factors down to the sites registered to the VO GIIS.

Each site will require an LDAP service certificate that should be provided by the RA. The Service LDAP cert in effect causes MDS to run queries as another account. The introduction of authentication and authenticated binding between GRIS and GIIS is liable to cause a slowdown in query times. This will be tested out between Grid3dev sites and the Grid3dev Top level GIIS and reproduced on the Grid3 deployment at the time of upgrade. An upgrade plan will be cooked up once a schedule is in place.

**VDT 1.2.0:** The VDT toolkit 1.2.0 has adopted all the changes into its MDS package and with latest tuned Grid3 configuration settings.

**Site upgrade:** The site upgrade plan for MDS will be carried out first in Grid3dev before moving over to the main production Grid3. The procedures and configuration settings will be determined in Grid3dev this is now possible because there will not be the impossible task of mimicking VO GIISes and their intricacies.

**Hand tuning:** Hand tuning of GRIS and GIIS query return times is the only method to determine values for a comfortable MDS deployment. Queries are timed, important queries between sites and between Top level GIIS

**Cascading query interval script:** The practice of running this script is to make it iterate around the overall timeout set value for the Top level GIIS. Currently it is being run once every 180s (3mins).

Correlation of MDS information and GridCat provide information consistency.

### 2.5.1. Program of Work

#### Deployment plan

Grid3dev Grid: VDT 1.2 will initially be deployed on Grid3dev sites. The software is available and it is expected that a majority of the sites have actually installed the software. Site administrators should have applied for LDAP service certificates, the

certificates will be needed for running basic resource queries across the deployed Grid. Variables for having a seamless information service across Grid3 should be solidified in Grid3dev, the eradication of one level of information indexing should in turn eradicate the need for more adjustments when Grid3 migrates to the latest software.

### **Monitoring of MDS infrastructure**

A package of monitoring scripts will be run at ISI in addition to the IGOE activities and websites. The Idea is to monitor the number of elements and resources being published by MDS across the Grid and to notify sites in the event they fall or information is compromised due to any reason including incorrect configuration.

### **Hawkeye integration**

In addition to MDS we will be unleashing a brand new monitoring user interface for Grid3 sites that will in addition to a myriad of things utilize Condor Hawkeye data. Globus will be collecting information from Hawkeye modules running on Grid3 sites and begin archiving them in an attempt to provide a historical perspective to Grid monitoring. A vast majority of Grid3/Grid3dev sites use condor and know about Hawkeye, the Hawkeye modules should be installed so as to have this data available. There is more on this in the New MDS Systems portion of this document.

## **2.6. Site Verify**

Site verify scripts are...

### **2.6.1. Program of Work**

Near term issues with site\_verify.pl include:

- o extracting expiration date of remote host certificates
- o add tests for batch job submission and cancellation
- o report paths to remote queue commands
- o report load average on remote gatekeeper
- o Handle case where MonALISA is not running on the remote gatekeeper, but on some other machine
- o Restore tabular report format when multiple gatekeepers are checked
- o Update documentation
- o Logging of test results to an XML file, or similar, so that an application may process the results.
- o Restoration of MDS tests, with an option to control whether or not tests will use MDS or not

Longer term:

- o It would be easy for site admins to impose reasonable security permissions (i.e., permissions, etc.) that could defeat some of site\_verify's tests as they are now implemented.

## 2.7. VDT Issues

### 2.7.1. MonaLisa version update in VDT

Currently, the MonaLISA software can perform a self-update (by retrieving and installing a newer version from a specified URL). This update can be initiated either automatically (by enabling the auto-update feature in the MonaLISA configuration file) or manually (by invoking an appropriate script).

This functionality is currently not recommended at Grid3 sites and should be disabled in the Grid3/VDT MonaLISA installation. The main reason for this is to avoid haphazard updates and version incompatibilities that might ensue. Instead the upgrade of MonaLISA should be dictated by Grid3 coordinators. When a new version of MonaLISA becomes available, the Grid3 coordinators will review its new features, and determine suitability for Grid3 and compatibility with older versions. Depending on how urgently the new version is desired, Grid3 will either issue an advisory for a coordinated Grid3-wide upgrade or decide to put it off until the release of the next version of the VDT (which will contain the new MonaLISA version).

Starting in VDT 1.2.1, VDT will change the MonaLISA configuration to NOT publish information under “grid3” monitor group (*lia.Monitor.group* in *ml.properties*) by default, in order to prevent non-Grid3 VDT sites from erroneously being reported as part of Grid3.

### 2.7.2. Configuration in VDT

VDT 1.2.0 and above includes Globus MDS.

By default, MDS (GRIS, GIIS and various information provider scripts) is configured to run as a non-root (*daemon*) user, to minimize the potential impact in case MDS scripts are compromised.

MDS will be configured to only accept authenticated (not anonymous) queries.

MDS will require a separate LDAP GSI service credentials (cert and private key) owned by the *daemon* user. These will be distinct from the (root-owned) host credentials. The credentials will not be installed by default as part of VDT/Grid3, but need to be obtained after the installation, by proper certificate request to one of the recognized certificate authorities. All GIIS hosts for a VO will need to be manually configured to recognize and accept all other GRIS/GIIS service certificates.

## 3. Service Health and Troubleshooting

### 3.1. *GridCat*

Front-end will become more important in longer term. This is because some of the duplicate tasks performed by the GridCat might be transferred to other monitoring tools in this time frame. It might form a thicker layer than what it is now.

Front-end should provide a better looking status map with better scalability, possibly with the application of java script like format. It should have more flexible presentation method than now. This could be accomplished by using the XML data model or templated data so that it can present the catalog in more versatile forms.

One particular area to provide more flexibility is adding the editable attribute to the Front-end. The editable attribute should be elaborated with the capability of editing by the site admin or the appropriate person. This can be achieved by providing the secure web server with grid-proxy based access to such kind of editables.

On the other hand, Back-end side of the Grid tests should become more granular than now by adding more and useful tests. In order to maintain consistency between the GridCat and monitoring tools, a way of establishing communication with monitoring tools, MDS, MonAlisa, and Ganglia should be provided. This will provide a way of consistency check of the GridCat itself.

In a more detailed technical side, all the scripts should be made modular and object oriented implementation of the scripts so that they can be reused. Also in order to cope with the portal like access to some editable attributes, better organized database schemas should be carefully decided. If it is necessary, we will rewrite the schemas for the database. Also for a better or versatile presentation of the catalog, XML can be employed and We can store information to the XML and MySQL DB.

### 3.2. *Grid Exerciser*

#### Summary

The Grid Exerciser (GEx) maintains a continuous load of simple jobs on one or more sites. It can generate reports on throughput and errors encountered.

#### Present uses

Assuming that various sites are configured to give the GEx jobs absolute lowest priority, the GEx acts a measure of unused processor time. Notably, since the GEx is running actual jobs, the unused processor time is known good.

The Grid Exerciser acts as a continuous real-world test of site functionality. Sites may not be used for a period for any number of reasons; the GEx ensures that they remain functional during those lulls. Furthermore, because the GEx tests a moderately deep, realistic code path, it can reveal incorrect behavior. not revealed by more limited use cases. For example, several Grid3 sites suffered from a bug present in a VDT release that

broke file staging. No users of those sites were yet using file staging, so they had not noticed. The Grid Exerciser ran into problems with these sites, causing the bug to be noticed, diagnosed, and fixed.

## Error Handling

In its place as a real-world test, the GEx uncovers problems, some transient, some permanent. At the moment Alan De Smet investigates the problems as time permits, informing the iGOC and remote site admins with initial assessments. In the initial investigations, tools like SiteVerify are useful.

At the moment there is no established procedure for handling problems reported by the GEx. To provide useful reports to the iGOC, the GEx maintainer needs to examine local GEx related log files, as well as remote sites. This would complicate directly handing GEx monitoring to a third party as they would need easy access to these log files.

## Future Uses

By submitting a steady stream, the GEx can encounter scalability problems before other projects using smaller subsets of the grid. To this end, the Grid3 GEx deployment is slowly being scaled up.

Summing the daily CPU hours used by GEx on Grid3 with the daily CPU hours used by other users, all CPU hours potentially available should be accounted for. Some error is to be expected due to overhead in the system; this total will make it possible to quantify the actual overhead. If the overhead is unexpectedly and undesirably large, this will reveal that problem. Unfortunately, this particular test requires that the GEx submit enough jobs to saturate Grid3; in light of expected scalability problems, this is further in the future.

## Questions:

- Main purpose is to do scalability tests and find problems before applications do.

- Discussion about filling grid to capacity – a principle that a resource is best characterized by fully utilizing it.

- More for integration / development than for production environment.

- Problem hangs at UB. Mark and Alan to deal with offline.

- Include as part of responsibility reporting of the problems to appropriate parties?

- Grid monitor running?

- Need a formal process for handling the errors.

- More specific about the errors?

- What is the application doing itself?

- What is the roadmap for the application and program of work?

Does it impact systems, like schedulers?

Concern for preemption. Alan will send email to grid3-admins.

### 3.3. GridScout

As one of the outcomes of the Grid3 monitoring meeting, an idea for a new software component has been proposed. The software (temporarily named “Grid Scout”) will be designed to diagnose/predict common configuration/run-time problems at grid sites. Such common problems may include:

- Invalid GSI configuration
- GASS cache / scratch space filling up
- MDS information provider malfunctions
- Unreasonably high CPU load, etc

Among other things, the software will know how to extract information from logs for Gatekeeper, GridFTP, batch systems, diagnose batch system installations, identify orphan files and processes, etc.

It is currently not determined when and how the software will be invoked (via GRAM submission, SSH, manually, etc).

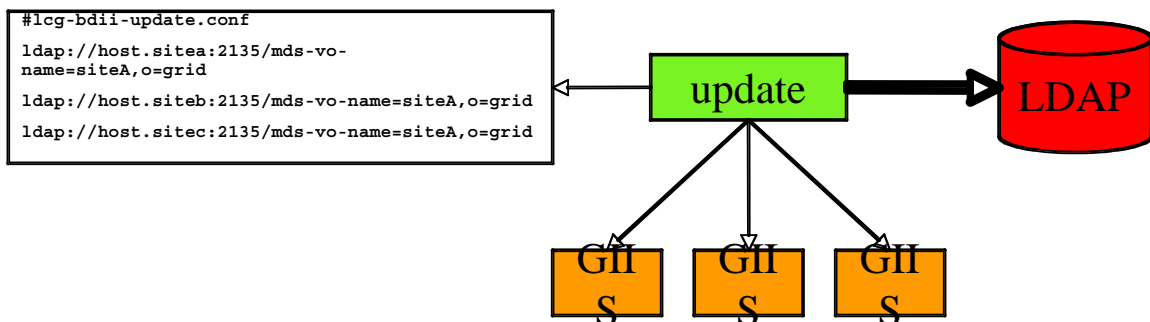
The primary responsible for the software would be the VDT team, but they may incorporate contributions from other projects.

## 4. Information Systems

### 4.1. Interoperability with LCG

Grid3 has been approached by LCG to solve the interoperability issues between Grid3 and LCG information providers. This is a necessary step toward a complete set of interoperable grid services in the medium term, while in the short term it is required to allow the EDG resource broker to submit any jobs to US Grid3 sites.

LCG uses an information provider architecture based on BDII, which is flatter than the



architecture currently used in Grid3. A service polls the site GIIS services regularly and updates a database. Resource brokers and other services requiring up-to-date

information can query this database. LCG believes the architecture, shown in the figure below, can scale to 1000 sites. The service that queries the individual sites could be configured to filter on the site information, allowing heterogeneous information services to feed into the LCG information store.

While this technical solution to different information systems provides considerable flexibility, Grid3 and LCG should assess carefully what differences are actually required by both grid projects. Grid3 is in the unique situation that we are not extremely dependent on the information services for the efficient functioning of the grid infrastructure, so now is an opportune time to reassess the required differences. Grid3 should be flexible with respect to changes and approach the interoperability effort with the goal of reaching transparency.

Both Grid3 and LCG currently use the same version of the GLUE schema (v 1.1), but there are differences in the information published into the schema. There are three information provider issues that present themselves immediately, though there are likely to be additional more subtle issues as a more thorough investigation is performed. The first is the Grid3 and LCG use different information provider and Grid3 ones seem to be outdated. Furthermore Grid3 has not been as thorough about the information provided. One item on the path to interoperability is to ensure that all Grid3 sites are consistently reporting the fields in the Glue schema. The second issue is the differences between extensions to the information providers. Grid3 has loaded a number of site environment information into an extension of the schema. Some of the information is duplicated in the form of LCG environment variables. While this is reconcilable, it does require negotiation or at the very least consistent duplication. The final immediate issue was the publication of authorized users from the gridmap file into the information provider. LCG but not Grid3 use this technique. This is again an area that needs evaluation by both sides.

#### **4.1.1. Program of Work**

Grid3 should identify effort to perform a number of small evaluation tasks over the period of the next month and a half.

- Grid3 should provide a test installation with a fully deployed version of the updated information providers
- Grid3 should survey what is collected and published currently in the GLUE schema, what will be published by the updated information providers and what would be required to make it more consistent with LCG collections
- Grid3 should examine the similarities and differences between the GLUE schema extensions adopted by Grid3 and the site environment parameters used by LCG.

The goal of this effort should be to arrive at an information provider system that is interoperable between the two grid efforts and does not require translation by each group. If we do have to implement filtering, the need for the differences should be justified and understood.

The Grid3-dev testbed is a logical candidate to test and verify new information system configurations and ensure that the grid monitoring continues to function as before. Grid3 should prepare to deploy a revised information system on some fraction of the Grid3-dev sites by the end of August. This could be used by both the iGOC and LCG to verify it meets the needs of each group.

One issue that was raised during the creation of the original GLUE schema was the need to be able to upgrade and extend the information system in a timely manner. This has never been achieved in practice. Grid3 should consider making the information configuration an area of the cache that could be patched and updated without a full formal release of Grid3. This would hopefully allow more rapid deployment of extended information infrastructure. The same deployment infrastructure that might allow Mona Lisa upgrades to be fully described by a patch number on top of the Grid3 release would be required for information systems.

## **4.2. New MDS Systems**

Grid3 runs MDS2; MDS3 is latest version; MDS4 is under development (scheduled for ~Feb 2005). Protocols & data model: MDS2 uses LDAP; MDS3 and MDS4 use XML over (resp.). OGSi and WSRF protocols, both of which are built on webservices.

There is no more single GRIS per resources: OGSi and WSRF services (such as GRAM) publish their own monitoring and discovery data; however, it is still possible to run services to host arbitrary information providers (which now must output XML, rather than LDIF).

Service-wise, we still have index (GIIS in MDS2, but we don't call it that any more); under development are an archiver, a trigger service (that will, for example, send mail on fault conditions and other defined events), a web UI and more integration with other information sources, Ganglia to begin with, although we are also doing some prototype work with Hawkeye. These are gatewayed to publish their data in the GLUE schema (CE only, as we traditionally have been focused around reporting GRAM information).

We are also making more effort to make this stuff deployable out of the box, with better documentation on how to tie it all together and tune it.

### **4.2.1. Program of Work**

Specific steps in either Grid3 or OSG-0 context...

## **4.3. Use Policy Information**

Usage policies (UP) issues arise at multiple levels when resources are shared. In the scenario where resource owners want to grant to different virtual organizations (VOs) the right to use their resources, owners want to express and enforce different policies under which resources are made available. We identify two levels where UPs are handled: site-level and grid-level. There are at each level several individuals and components that handle UPs. At the site level, resource owners state how their resources must be allocated

and used by different VOs. This represents the *high-level GOAL a site's owner wants to achieve or manager policy (MP)*. The site administrators map MPs to different software RMs' syntaxes. The end product is the set of RM configuration files, named *the local POLICY or system configuration (SC)*. At the grid level, UPs are translated from SC by automated tools into an *abstract policy (AP)* set. SC descriptions are collected from the site RM configurations, filtered and, after translation, published through any standard Grid monitoring system, e.g., GridCat, MonaLisa or ACDC Monitoring Tool.

To give fast intuition, we consider in this document only what a site does from the perspective of a particular VO, e.g., "*site X gives ATLAS 30% over a month*". Thus, we have identified as important for the monitoring work the SC description at the site level:  $SC(VO) = \langle \text{number of nodes, scheduler-type, scheduler-config} \rangle$ , which is written by the site administrators during RM configuration process. Such SC descriptions are collected and translated into  $AP(VO, \text{Site})$  descriptions, which expresses  $SC(VO)$  in some more abstract, a scheduler-independent format. As an example, the VO settings at different sites are presented in the following table.

Site Name	# of CPUs	Allocations per VO (in %)						
		iVDGL	USATLAS	LIGO	SDSS	BTeV	GridEx	USCMS
t2cms0.sdsc.edu	76	0.62	24.74	-	24.74	24.74	0.00	0.40
nest.phys.uwm.edu	305	0.00	7.28	0.00	0.00	0.00	-	0.00
uscmsb0.ucsd.edu	3	11.68	11.68	-	-	-	11.68	11.68
xena.hamptonu.edu	1	25.00	25.00	-	-	-	-	25.00
garlic.hep.wisc.edu	101	3.01	3.01	3.01	3.01	3.01	-	3.01

The ultimate goal for the UP monitoring is to achieve the exact status of running jobs for a VO at a site. In the next table we capture (in terms of jobs slots and jobs) our view of assessing how a site achieves the policies it wants. Target represents the number of allocated slots for a VO, Current represents the number of actual jobs running, while Demand represents the queued jobs at the site. The last column is the actual assessment of UP violations: for example, USATLAS is under its allocation with jobs waiting in the queue, which is a clear violation of the site UP, while LIGO is under its allocation only because not enough jobs are ready for execution.

VO	Target	Current	Demand	Level
USCMS	60	50	50	OK
USATLAS	20	15	30	Under
bTEV	10	10	100	OK
L I G O	5	3	3	OK

SDSS	5	22	50	Over
------	---	----	----	------

All these metrics are further important for policy-based scheduling decisions. We envisage that planners, work-runners, or even applications will invoke VO decision maker tools (V-PEPs) to get a recommended execution site for each job. Such recommendations have to *policy-cognizant* to achieve better Grid utilization and user response times (both site and VO policies). A V-PEP provides answers based on site policies, VO policies, job requirements, sampled monitoring data, and gets queue conditions, local site policy from monitoring system; gets job resource requirements from any monitoring tools that observe usage policies at sites (S-POP). There can be one V-PEP per VO or one V-PEP per Grid, also V-PEPs can be multi-layer (Grid, VO, and Group). Such issues are further explored by authors.

#### 4.3.1. Program of Work

Site Policy Observation Point (S-POP) already runs on Grid3 and all raw data are available through http or from a postgres database. The VO policy enforcement point (V-PEP or UP site selector) is under development as an OGSi service. It is composed of a site selector, a site recommender and a site predictor. So far, the UP site recommender is already available and it can be accessed thru specialized clients which are already available. The site UP site selector is in progress and it should be ready before the Super Computing '04. The framework we envisaged so far is composed of one V-PEP for the entire Grid3, but we expect to migrate soon to a solution with at least one V-PEP per VO. There are also some improvements needed for S-POPs in terms of site RMs it is able to query.

## 5. Grid2003 Lessons Coverage

Lots of good stuff - We can make use of Computing services at Grid3 sites, we are on the verge of reliable MDS deployment, local autonomy

works well, etc.

But,

- o We still do not have disk management services, though some site administrators have policies in place
- o It is still hard to debug 'why jobs fail' - though some/most of this practically is best left as the responsibility of the VO/application admin.
- o Not sure if resolved: gass\_cache in AFS space, jobmanager patches solve stage out issues (?)
- o Expression of site policies (job execution, \$DATA/\$TMP cleanup, etc.) not yet widely/consistently done
- o No efficiency metric (% of free cycles used) tracked

o Head node load still can get high - it would still be handy to have

Some recommendations for minimum hardware and software services. o Reliance on shared filesystem still exists o LCG-2 Interoperability should be pursued.

## **6. Observations on OSG-0**

Data grid monitoring information is rapidly moving to the forefront with respect to efficiently utilizing the computational grid resources. The availability and reliability of the existing grid enabled data repositories is vital in achieving the high-performance computational grid throughput desired by the current Grid3 application managers and future OSG-0 participants. The core monitoring efforts currently presenting in the Grid3 cyber-infrastructure deployed is doing a reasonable job of monitoring the computational grid jobs and gatekeeper performance and availability. Although additional work is required in this area application managers can easily obtain enough information to successfully run their production jobs at a reasonably high throughput. The current level of monitoring needs to provide the corresponding network bandwidth and latency estimates for staging in and out the required data files for all computational grid jobs. We have at best provided rudimentary measures and tools for the monitoring and reporting of this information. Additionally, the historical measurements of the Grid3 data grid transfer rates should serve as a base for estimating future OSG-0 participant utilization and scientific application profiling. Providing a service based data grid monitoring cyber-infrastructure will facilitate predictive and intelligent scheduling of computational grid jobs based on data locality, network performance, computational resource availability, data grid repository capacity, quality of service and local site Service Level Agreements and policy.

### **6.1. *Concepts and Principles***

### **6.2. *Development/Integration Grid Suggestions for OSG***