GWD-??
Grid User Services Research Group

Jim Giuliani
Ohio Supercomputer Center

February 2003
Revised June 2003

Support Services and Tools Requirements

Status of this Draft

This draft invites discussions and suggestions for improvements. The distribution of the document is unlimited.

Copyright Notice

1 Abstract

As Grid based computing develops as a regular mechanism for use by researchers, many issues arise in the context of supporting the applications and their developers and users. This document is intended to provide information to grid infrastructure developers and operators about the requirements of those providing that support to be effective in their role. It is well known that those considering developing and using grid-based applications for their research will have a strongly negative reaction if they have a poor experience when they need support in their endeavors. Here we outline a number of grid services, tools and capabilities that grid applications and grid user support staff require to be responsive to those needs.

Table of Contents

2 Introduction

As computing centers enable Grids for application development, user support for these computational Grids will most likely fall to existing User Service Support groups. While many current software tools and diagnosis techniques can be applied, Grid applications by their definition will require additional tools and analysis techniques to assist the Grid user community develop, debug and run applications.

This document is intended to provide guidelines to Grid infrastructure developers and support staff centers regarding the information tools and diagnostic applications that will be required to assist the user community in utilizing this new computational framework. This document provides technical discussion and detail about the user service framework outlined in a larger GGF document, Grid User Services Common Practices. The Common Practices document surveys some of the current and planned practices in some developing distributed environments and suggest the best practices as appropriate for various elements of the stated support model. The intent is to provide recommendations as to how to best support users and applications in these nascent environments.

As Grid environments and software configurations develop, it is recognized that additional services, information and tools may be required to enable applications and their support. This document will outline a base set of requirements that each application should be able to meet without significant difficulty. Periodic review and updating of this document is expected, but base functionality should not change.

3 User Support Issues

A document that claims to define the tools needed to effectively support users of a computational Grid would be negligent if it did not first describe the consulting and support issues that drive the need for these services.

This section will describe the user support environment in which these tools will be utilized. Since development of support tools follows the environment, as the Grid evolves any changes to the framework described below will most likely require refinement in the tools.

3.1 Single Resource vs. Multiple Resources

The first shift in thought that must be taken is that the diagnosis of a problem can and most likely will involve multiple grid resources. For the scope of this document, a single grid resource is defined as a machine or collection of machines that share a common configuration, like software and network file systems, and appear to the user as a single system. A cluster of SMP machines is a primary example, where a collection of individual machines are united by an internal network and batch or scheduling system to run serial or parallel user applications. In this configuration a single software install is often shared among machines via a shared file system.

In a single resource, system services and capabilities will read similar, if not the same, binary and configuration files. Applications that run on a single resource, therefore, will generally not run into incompatibilities between software versions. In a Grid environment where virtual organizations may encompass multiple systems under different administration groups, incompatibilities in software are likely to be a key issue.

Grid applications that span multiple resources now yield multiple service log files to examine. A job executing across several grid resources presents the support staff individual with multiple grid services log files, any one of which may hold the critical error message that identifies the cause of a failed job.

It is important to understand that not all Grid applications will utilize multiple resources. Taking an existing application and using the Grid interface as a means of submission to a batch queue of a specific machine or cluster is a first step that will be taken by many application developers.

3.2 Live Application Analysis vs. Post-mortem Analysis

The shift to a Grid computing environment also implies a shift in the primary roll of the User Services support staff. Problems with current single resource applications often are presented to the User Support staff with the implication of finding the exact location of a failure in an application.

The primary role of User Services in a Grid environment will be to identify the general area of failure, which application and/or which grid resource, rather than the actual location of the failure within a specific application. Determining the failure within the application, i.e. code debugging, will be a secondary role of the Grid support individual. This might be possible for very small or simple grid applications, but most grid applications will be too complex and beyond the scope of someone outside of the development program to resolve. Determining if an application failure was the result of a system problem or a user application failure will be the initial problem diagnosis. Identification of a system problem would then involve systems staff to resolve the issue. Problems with a user application can be isolated to the given application and pertinent error messages highlighted, but the user community should then have to debug their code.

Based on this idea, most of the tools outlined in this document focus on probing and discovery of the application that failed and if the failure was due to the application or the environment.

3.3 Scope of this Document

The issues that now differentiate grid applications from traditional single system applications are:

- Possible multiple job managers, resulting in multiple log files that must be examined
- Possible multiple standard output/standard error
- Correct software configuration on one system and incorrect software configuration on another as each could have different configurations or versions of software

These are the general catalysts for developing the changes outlined in this document. Consulting issues are covered in more depth in the Grid User Services Common Practices document.

Scope of this document:

- Scope of job
    - Single grid resource utilized
        - Take existing job that is submitted via batch to a single system and use Grid interface to submit
    - Multiple grid resources utilized
        - Co-scheduling
        - Multiple systems and multiple log files

4 Grid Services Requirements

There is a set of information that is important for users to know about in order to target the resources they wish to use. Frequently users can make use of various resources to accomplish the task they have at hand, but need the ability to decide which resources they will use. With evolving Grid applications such as portals, dynamic allocation of resources and resource brokers, users may not know the specific hardware their applications run on.

It is therefore important for support staff to have access to this information. This allows the staff to offer assistance to users in selecting resources, but also allows them to assist in determining what has gone wrong when there is a problem with the execution of a task or set of tasks. With possibly large quantities of resources being allocated to jobs, this information also needs to be assembled automatically by user support tools.

Much information about Grid resources is published through the directory services of Grid environments and accessible through APIs. Web interfaces and stand-alone applications have been developed to allow easy browsing and searching of directory service data. An example is the MDS browser developed for the Globus Toolkit (see http://www.globus.org/mds/getmdsdata/vo-index.html for examples).

In addition to published information, information is available though Grid service log files. Since there are multiple instances of Grid services, multiple job managers for example, it will be necessary that log files be kept in a known location or that their location be published so that tools can find the information and parse it to extract out details pertaining to a given Grid application. A mechanism must also be put in place to allow for consultants to view log information and standard output/standard error streams.

4.1 Job manager logs

Even in future Grid environments, a job manager service will be required on any computational resource to enforce the use policy established by the local organization. Currently users interact with the job manager via command line functions to submit and query a batch scheduling system.

This interaction between user and scheduler may be abstracted out by resource brokers or portals, but the job manager will continue to monitor and control job execution on local hardware.

Since job managers are central to the execution of any Grid application, the log files generated by the software are of significant interest and importance to user support staff trying to investigate the failure of an application. A log reporting service would be desirable as a component in the jobmanager software. Applications would then be able to access and query the log information through an API. The specific details of this are unclear at the present time.

At present, the recommended method of gaining access to jobmanager logs is to publish the information in the directory service. An example using the Globus Toolkit configuration would be as follows:

```
Mds-Software-deployment=jobmanager-instance
        ObjectClass = ServerLog
        ObjectClass = SchedLog
        ObjectClass = ExecuteLog
```

Local system policies need to make provisions for the log files to be readable by one of the following methods:

- A Limited Super User (LSU) or SUDO utility, which is described in section 4.2.
- Group access permissions for a local Grid consultant account, which is described in section 6.1
- Readable by any account on the system

With access enabled, application tools can fetch the log files from the remote resources. Information can be read dynamically from the Grid resource or cached locally. The information and applications to process the log information should provide the following functionality:

- Ability to view log file for a minimum of 2 weeks without local staff intervention
- Ability to apply filters based on job id, user id or date/time
- Ability to select windows around entries tagged by searches
- In the event of a Grid application that spans multiple resources, the ability to examine, search and tag events from multiple job manager logs simultaneously

4.2 Grid Enabled SUDO (Super User DO)

A tool that is prevalent in existing support centers is the Limited Super User (LSU) or SUDO function. sudo is a program that allows system administrative policy to give certain users the ability to run some (or all) commands with root privileges.

As with other utilities that have migrated to the Grid environment, like Grid enabled ssh, the Grid enabled sudo should use a valid proxy for authentication and if authorized, will allow certain functions to be performed on the Grid resource.

Local organization policy will dictate what services, if any, are enabled. It is assumed for the scope of this document that user support staff, while they will have access to the systems and environments they support, will not have super user privileges. While in some environments such access might be the case, in general, it cannot be assumed.

4.3 Published Grid Resource Status

It will be discussed in section 5 that a tool should be developed which, when given a list of resources via a grid service job id or range of ids, fetches all of the resources that have been allocated and then builds a text or graphical display showing status of every resource.

Fundamental to this application is having the status and performance data needed to analyze system functionality and performance issues. Currently a subset of dynamic system performance data is available through the MDS tree. An example from a system configured with the Globus Toolkit is:

```
Mds-Device-Group-name=processors
        Mds-Cpu-Free-1minX100
        Mds-Cpu-Free-5minX100
        Mds-Cpu-Free-15minX100
```

The information that is currently published is very limited. There is a wealth of static, quasi-static and dynamic data that will need to be published to allow support staff to efficiently browse and identify resources allocated to an application. For quasi-static and dynamic parameters, an update frequency will also need to be specified, so that the values are updated on a regular interval. A preliminary, by no means complete list is:

Static parameters:

- Peak network bandwidth of each network interface

Quasi-static parameters (24 hour time scale for updates):

- System utilization
- System uptime

Dynamic parameters:

- Uptime

In this classification, dynamic parameters may have too high an update frequency for storage in a directory service. Web pages may offer a better means for collecting and archiving this information, with the web URL stored in the directory service. (should examples be given here, like the Ganglia Cluster Toolkit http://ganglia.sourceforge.net)

4.4 Scheduler Service Information

Current single resource scheduling technology covers a broad range, from simple first in first out (FIFO) policies to advanced scheduling policies, which optimize resource utilization and minimize job response time. Quality of Service (QOS) features are also available in more advanced software packages. These allow directed delivery of resources and services, policy exemption, and controlled access to special features.

As complex as the latter configurations are becoming, the Grid promises to add complexity with the ability to co-schedule multiple resources across organizations with possibly different scheduling policies.

Current scheduling software is queried via command line utilities, some of which require group or sudo privileges. Access to these through a consultant account or Grid enabled sudo will provide some functionality, but will not meet the full needs of the consultant staff. Scheduling software and co-allocating software will need to provide and API or some other hook to allow client applications to query multiple resources. This functionality will also make possible the opportunity to determine when the resources requested for a job will be available. This will be useful by both end users and support staff personnel.

A partial list of issues that might need to be identified by support staff are:

- Conflicts in user requests
- Conflicts between site use policies
- Potential corruption or failure of scheduling daemons

One initial effort will be to publish the location of log files, as discussed in section 4.1. This will help identify possible problems with the software and post-mortem analysis of failed jobs.

4.5 Access and Use Policies

It is important for the organization or collection of organizations providing the grid environment to appropriately set the shared expectations for the users of these environments and for those providing support. These documents will be of specific interest to user support staff trying to understand why a Grid application cannot be executed.

Since future Grids will be comprised of large amounts of hardware resources and span multiple physical organizations, these documents need to be installed somewhere accessible and easily referenced.

The World Wide Web is the obvious channel for delivering the access and use policies, as these documents should be publicly available. Section 3.2.7 of the Grid Constitution outlines additional non-performance information.

4.6 Software Installation

When software installation occurs, it should publish its information in the directory service. This should be integrated into the software installation processes by the developers of each package.

5 Tools

At this time, several tools have been identified as useful to the Grid user support staff and take advantage of the Grid services and information outlined in the previous section.

5.1 Matrix of Test Applications

One of the most fundamental problems that may arise when working in a distributed grid environment is that the environment between multiple resources are not configured properly to support a users application. This encompasses different software versions, OS or file system configurations, security settings, firewall or network configurations and local user account privileges.

A test suite should be developed that contains a small sample application for each Grid service. Critical also is an interface that would allow support staff and eventually users to build a larger test suite by selecting from the list of individual Grid service test applications. For each grid service selected, the interface will fetch a test application for that service and then assemble one large suite that encompasses all of the services selected. Since standardized test applications will be used as the building blocks, this will generate a script that tests the underlying framework of the Grid environment, eliminating the complexity of the user application. Obviously, a user application will not function if there is a problem with the underlying grid services.

Grid service developers will need to provide a small test application that verifies the behavior of their service. An application and a script should be provided so that they can be combined into the complete test application. The script and application should be self-sufficient and generate any files or data structures needed to test the service.

This can diagnose problems that range from expired certificates to a particular Grid service not being configured properly on one resource.

5.2 Service Log File Collector/Browser

As discussed in section 3.2, Grid application support will primarily be a post-mortem analysis. The majority of Grid applications, due to their asynchronous execution, distributed execution and complexity will make it impractical to “gather” the user code and run it. In addition, resources that a job may have been allocated for one run could differ drastically from the resources allocated for a second run.

All pertinent log files from all grid resources will need to be analyzed. Entire logs, usually encompassing an entire day may be too large to practically browse. Extracting entries keyed to a specific user name or job would provide a concise summary, but general log entries that might be important would be missed. The application should therefore offer a choice of fetching the entire log files or entries keyed by user name or job.

Ideally, Grid services will provide some form of remote access to log information. In addition to this, Grid service developers should publish log file information in the directory information for the service. As an example - job manager services, such as PBS and LSF, should publish locations for both the batch server and the scheduler logs. Additional services, such as the MOM process logs for PBS batch system should also be published. Some additional knowledge will need to be published, as sometimes the log files will not be on the grid service machines, so hostnames may also be needed. An alternative would be to require that a machine be configured as a log server, where log files from all machines would be collected, stored and referenced.

5.3 Standard Output and Error Collection

(this section needs to be expanded)

For most post-mortem analysis, job standard output and standard error will have been returned to the user. One sub-class of support problem is to determine the status of a currently running application. For this type of problem, collecting the standard output and standard error are important. This can also be an important diagnosis tool for users.

Job managers should direct standard output and standard error streams to the GASS cache rather than static files on the local systems. By directing the streams to GASS cache, it will permit the output to be access by Grid utilities during a run.

Additional functionality might be to have a residency time for the error and output streams in the GASS cache. This would permit support staff doing a post-mortem analysis to have access to the output streams for some time window after the run has terminated. This has implementation and security issues that have not been addressed yet.

5.4 Grid Debugger

An application debugger is critical to any support function. Due to potential high network latencies between the client machine and the remote resource where the application is running, current debuggers that launch graphical displays from the remote resource are not adequate.

A client-based structure should allow for remote debugging of Grid applications. The local graphical process can communicate with the remote debugging application, minimizing network communication.

The scope of the debugger has not been determined yet. Should the scope be limited to looking at only a single Grid application (a serial program, OpenMP parallel program or MPI parallel program) or have the ability to look at multiple applications that do not have a tight binding, like that which MPI provides.

5.5 Directory Service Browser

There are already multiple directory service browsers available. Additional functionality is needed for the user support staff.

In addition to starting at some point in the directory tree and expanding down, the ability to select a subset of machines in the tree is required. This should include the ability to select machines, via a check box, from a complete list as a filter for future searches, or by a job id.

Browsers should also recognize http tags so that web links published in the grid directory service could trigger the directory browser to launch a web browser to display access and use policies, as discussed in section 4.5.

5.6 Grid Resource Browser

The primary function of this tool is to provide a snapshot of a group of Grid resources and their connections. As discussed in section 4.3, there are additional static, quasi-static and dynamic information that should be monitored and published for each system.

This utility should be developed such that resources can be manually selected, given a Grid job id or range of ids. It will then fetch all of the resources that have been allocated, build a graphical display showing status of every resource and interconnect status. This display can then go through MDS or a web page to find things like load history,etc.

There is additional dynamic information that can only be measured by running an application. Included in the functionality of this tool should be the ability to launch diagnostic applications. One example is an application to measure bandwidth and latency parameters between two Grid resources.

6 Capabilities

A certain level of access will be needed by support staff personal to get access to system logs and status information. This will need to be addressed in the User Services Agreement documents.

6.1 Consultant accounts

The general functionality is that a consultant account be generated on each system that falls under the support agreement. Support staff access can then be managed through grid mapfile entries and each system can control access of the User Services staff by configuring a single consultant account. If limited super user privileges are supported, only the consultant account need be authorized.

It is not envisioned that the ability for a consultant to enable or become a user will be needed. Most Grid applications will be complex enough that “becoming” the user to run the application will not be a practical diagnosis tool.

In situations where this is needed, data conferencing technology can be used to share desktops. This should provide enough functionality and is supported with existing technology.

The consultant account should have the following privileges:

- Access to a limited super user utility like sudo, if the policy allows

- Group privileges to access all Grid service and batch scheduling log files
- Ability to access batch scheduling system utilities

6.2 Knowledge base

6.3 Trouble ticket system

Waiting to see what the scope of the GUS Trouble Ticket Exchange document will be.

7 Summary/Conclusions

8 Security Considerations

Though this document does point out the need in various areas to define the security practices to be used in a particular Grid environment, it does not advocate the use of particular policies or technologies to implement those policies.

As discussed in section 6.1, the ability to enable or become a user is not seen as critical to the support staff role. Eliminating this feature eliminates numerous security considerations.

It has also been mentioned by some individuals that the default information published by the Globus MDS make too much information available and pose a security risk. This document calls for information above and beyond what the current default is. This raises the issue of an overall security framework. To participate in a Grid environment implies that grid resources make a great deal of system information available to the user community.

One possibility is to require a valid proxy to browse directory information. One other is to restrict sensitive information, such as source code, to client only machines. Most of the features discussed in this document involve interaction with Grid services, many of which do not run on Grid client installations. This affords the user increased security and privacy on the client machine.

9 Author Contact Information

James E. Giuliani
Ohio Supercomputer Center
1224 Kinnear Road
Columbus, OH 43212 USA
jimg@osc.edu
ph:+01-614-292-9291
fx: +01-614-292-7168

10 Intellectual Property Statement

The GGF takes no position regarding the validity or scope of any intellectual property or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; neither does it represent that it has made any effort to identify any such rights. Copies of claims of rights made available for publication and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this specification can be obtained from the GGF Secretariat.

The GGF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights which may cover technology that may be required to practice this recommendation. Please address the information to the GGF Executive Director.

11 Full Copyright Notice

This document and the information contained herein is provided on an "AS IS" basis and THE GLOBAL GRID FORUM DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE."