GWD-??
Grid User Services Research Group

Jim Giuliani
Ohio Supercomputer Center

February 2003
Revised June 2003
Revised September 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 applications.

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 Level of User Support

Grids will exist across a wide spectrum of organizational size and structure. The group tasked with providing technical support could range from a singe individual to a large group. In reality the user support structure will vary in size and methodology based on the individual needs of their user community. The level of support to be provided will also vary for each organization and should be defined by a service level agreement. Detailed information on service level agreements is outlined in chapter 3 of the Grid User Services Common Practices document, End-User Service Level Expectations.

For planning purposes, the tasks of the User Service Support group can be segregated into two categories, which are differentiated by the level of support given to the end user. The first category covers support from a group that is tasked with providing problem analysis and some high level application debugging to a large user base. This scenario falls in line with many of the larger User Support groups, or help desks, where the support individuals have detailed knowledge of the computing resources and programming models, but in general are not familiar with the specifics of the user applications. The primary goal is to identify the source of the failure and who would be responsible for addressing the problem. Possible categories include systems problems and OS configuration issues, middle ware issues and finally application failure.

The second category is an extension of the first category where the support staff have additional knowledge of the application and have some responsibility to help diagnose/debug specific applications. Here, support staff may be allocated to a research project or perhaps someone who is a full time member of the research staff is tasked with support.

3.2 Single Resource vs. Multiple Resources

A fundamental 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 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 executable 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 have the potential to be a significant problem.

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.

With all of the promises that Grid computing holds for distributed computing, 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.

3.3 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 applications often are presented to the User Support staff with the implication of finding the location of a failure in an application on a single computational resource.

A fundamental role of User Services in a Grid environment will be to identify the general cause of failure, which application and/or which grid resource failed. If a job failure is found to be the result of an application failure, the specific program at fault can be identified and determining the failure within the application, i.e. code debugging, will depend on the service level agreement established by the service provider. This might be possible for very small or simple grid applications, or organizations that have computational staff integrated with the software development teams, but many grid applications will be too complex and beyond the scope of someone outside of the development team to evaluate.

Based on this concept, 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. This is a common step for any support staff, regardless of the level of application support offered. Where appropriate, tools for debugging individual applications are discussed.

3.4 Scope of this Document

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

- The need by user support staff to examine applications running on systems not administered by their local organization
- The possibility of multiple job managers, resulting in multiple log files that must be examined
- Possible multiple standard output/standard error
- Possible different software versions or configurations between computational resources

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.

User Service Level agreements are also an important factor for grid applications. While not unique to grid computing, their use in grid computing environments offers more impact. The broader scope of grid computing opens the support staff to a wider range of technology and problems. Without a clear understanding of the scope of user services, users will set unrealistic expectations on the level of support.

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 important for support staff to have access to information about which resources have been allocated to a job, as well as details regarding hardware, software and network interconnects. This enables the staff to offer assistance to users in selecting resources, but also allows them to diagnose the source of a job failure and if the problem was due to a system problem, system software or a failure in the users application. With possibly large quantities of resources being allocated, this information needs to be collected and assembled automatically by user support tools.

For some existing Grid configurations, information about Grid resources is published through information services and accessible through APIs. A key component of Grid User Services will be enhancing the information published. Web interfaces and stand-alone applications can permit easy and efficient browsing and searching of information 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, diagnostic data is available though Grid service and system service log files. Since there are multiple instances of these 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. This will enable support utilities to 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

It is envisioned that in 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 on the local system 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 Grid applications, the log files generated are of significant interest and importance to user support staff trying to investigate the failure of an application. A log reporting service or API for remote access would be desirable as a component in the jobmanager software. Applications would then be able to access and query this information as needed.

In the absence of this functionality, to parse log files for job information it is critical to know the location of the log files. Two methods can be implemented to discover the location of these log files. The first is to develop utility programs that have the ability to search a generic system and find the locations of various log files. Second is to publish this information as part of the software installation. The first method has the advantage that less customization is required on the client resource side. The only requirement is that the account under which the utility program will run has privileges to execute and search in the system area. The negative aspect of this configuration is the extra processing and I/O resources that will be needed to run the repetitive searches could be significant.

The recommend method of gaining access to jobmanager logs is to publish the locations. The information services used to publish this data will vary based on the grid software installed, but in general the access mechanism should be well known and accessible by utility programs. As an example, consider a system running the Globus Toolkit for grid services. The Meta-computing Directory Service (MDS) configuration would be similar to:

```
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
- Global read permission

With access enabled, application tools can either fetch the log files from the remote resources or spawn off search algorithms on the remote resource to parse the log files and return the reduced data. Information can continue to be read dynamically from the Grid resource or cached locally. The system service logs and applications to process the information should provide the following functionality:

- Ability to view log files for a minimum of 2 weeks without local staff intervention
- Ability to apply filters based on job id, user id or date/time window
- Ability to apply subsequent search parameters on initial search results
- 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

In addition to the above guidelines, system services should log as much information as possible. This is generally achieved by selecting verbose logging. Many key pieces of information are only recorded at the verbose logging level.

4.2 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 generally provides command line utilities to query system scheduling, some of which require special group or super user 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 and may not be viable in grids which span multiple physical organizations. Scheduling software and co-allocating software will need to provide an 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.

4.3 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. Generally there is a subset of static/quasi-static system performance data that is currently available on most systems. As an example, for a system configured with the Globus Toolkit, this information is available through the MDS tree:

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

The information that is currently available is very limited. There is a wealth of static data that will need to be published and dynamic data that will need to be collected to allow support staff to efficiently browse and identify potential problems with resources allocated to an application. A preliminary, by no means complete list is:

Static parameters:

- Information services
- i.e. Peak network bandwidth of each network interface

Dynamic parameters:

- Provided by a service
- i.e. uptime

In this classification, dynamic parameters may have too high an update frequency for storage in a information service. Web pages may offer a better means for collecting and archiving this information, with the web URL stored in the directory service.

4.4 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.

4.5 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 some non-performance information. Some additional items that should be identified are:

- Responsible systems administrator for each resource
  - Name
  - Contact information
- Support desk, if available, that oversees each resource
  - Contact information
- Referencing Certificate Authority

4.6 Software Installation

When software installation occurs, it should automatically publish its information so that users and support staff can access this information. This should be integrated into the software installation processes by the developers of each package or, if utilized, by a package manager.

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 available 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 customized test suite of Grid services which their application utilizes, by selecting from the list of individual Grid service test applications. 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, which will be used by the user, 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 larger, customized test suites. The script and application should be self-sufficient and generate any files or data structures needed to test the service.

5.2 Service Log File Collector/Browser

As discussed in section 3.3, a significant function of Grid application support will be post-mortem analysis to determine the location and cause of an application failure. 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. As recommended in section 4.1, verbose logging needs to be enabled. These logs, possibly spanning multiple days, may be too large to browse manually. 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 log files keyed by user name, job id, or within a certain time window.

Additionally, logic should be incorporated into the application regarding the structure and content of the various service logs that it can process. This logic will assist user support staff to gather and assemble the key pieces of information needed to form the global view of a grid job. As an example, most job managers return some unique identifier to the user when a job is submitted. From this job identifier, key pieces of information such as start time, end time, etc. can be determined. A job ID should not be required, however, so the functionality should be incorporated to allow a user support staff member to start with some form of user identification and an approximate time window and discover the key job parameters from the log data.

Given a resource or range or set of resources, the location of log files can be obtained from the published information outlined in section 4.1. Ideally, Grid services will provide some form of remote access to log information, but at a minimum the data can be processed by an application on the local machine. Some services, such as the MOM process for the PBS batch system, do not exist on the grid resource. Some additional knowledge will need to be published and access mechanisms will need to be provided. An alternative would be to require that a machine running grid services 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

One class of support problem is to determine the status of a currently running application. For this type of problem, accessing the applications standard output and standard error are important. This can also be an important diagnosis tool for users to determine if a solution is progressing properly.

Job managers should provide Grid enabled access to standard output and standard error streams. By directing the streams to Grid file system, it will permit the output to be access by Grid utilities during a run.

For post-mortem analysis, it would be desirable to have the additional functionality to set a residency time for the error and output streams in the Grid environment. 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 and not need to request the files from the user.

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 Information 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 application can then call other utilities outlined in section 4.3 to collect performance and load statistics.

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

Section 7.1 of the Grid User Services Common Practices document outlines the information resources needed to assist in the determination and resolution of problems.

6.3 Trouble ticket system

Section 6.1 of the Grid User Services Common Practices document discusses the nature of user questions and problem reports in a Grid environment. This section does a good job of outlining the basic structure of tracking user problems and assigning staff to resolve problems.

7 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 is also likely that to participate in a Grid environment implies that grid resources make a great deal of system information available to the user community. Many systems administrators and security personelle could raise issues with the amount of information that needs to be published.

One possibility is to require a valid grid identity, regardless of the authentication mechanism, to browse system information. One other is to restrict sensitive information, such as source code, to machines which do not run grid services. 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.

8 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

9 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.

Explicit statements about IPR should not be included in the document, because including a specific claim implies that the claim is valid and that the listed claims are exhaustive. Once a document has been published as a final GFD there is no mechanism to effectively update the IPR information. Authors should instead provide the GGF secretariat with any explicit statements or potentially relevant claims.

10 Full Copyright Notice