Jim Giuliani
Ohio Supercomputer Center
August 2004

Support Services and Tools Requirements

Status of this Draft

The Grid User Services (GUS) research group has reached consensus through e-mail mailing list discussions and GGF meetings that this draft is ready to be submitted to the GGF editor for review. Effort has been made to prepare it to the GGF document track guidelines outlined in CFD-C.1. Discussion and suggestions for improvements are invited. The distribution of the document is unlimited.

Copyright Notice

1 Abstract

As Grid-based computing develops as a regular mechanism for research, many issues arise in the context of supporting the developers, users and their applications. This document provides information to Grid infrastructure developers and operators about the requirements of the user services staff providing support, to be effective in their role. 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 if they are 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 provides guidelines for Grid infrastructure developers and support staff centers regarding the information tools and diagnostic applications 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 the more encompassing GGF document, “Grid User Services Common Practices” [1]. The Common Practices document surveys some of the current and planned practices in some developing distributed environments and suggests the best practices as appropriate for various elements of the stated support model. The intention is to provide recommendations as to the best way to support users and applications in these nascent environments.

As Grid environments and software configurations develop, additional services, information and tools may be required to enable applications and their support. This document outlines 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 describes the user support environment in which these tools will be utilized.

Since a Grid binds together various electronic and computational resources, the application fields and user communities that will utilize these resources will vary drastically. Since support systems match the environment, there is no single template for a support structure. The details described in this section should apply to most Grid user support systems. As the Grid evolves, changes to the framework described below will most likely require refinement of 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 an individual to a large group. In reality, the user support structure will vary in size and methodology based on the needs of the 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 tasked with providing problem analysis and some high-level application debugging to a large user base. This scenario corresponds with many of the larger user support, or help, groups 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 is responsible for addressing the problem. Possible categories include systems problems and OS configuration issues, middleware issues and finally application failure.

The significant difference between the second category and the first is that the support staff have additional knowledge of the application and have some responsibility to help diagnose/debug specific applications. 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.

The implication is that there is a common set of tasks that all support staff must execute in order to diagnose application or system problems. The second category of consult staff will likely go beyond identifying the location and general cause of the problem (i.e., system or application failure) and help identify the specific changes that need to be made to correct the problem.

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 will be defined as a machine or collection of machines that shares a common configuration, like software and network file systems, and appears to the user as a single system. A cluster of SMP machines is a primary example, in which a collection of individual machines is 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 may be a significant problem.

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

3.3 Live Application Analysis vs. Post-mortem Analysis

The shift to a Grid computing environment also implies a shift in the primary analysis task of the user services support staff. Problems with current applications often are presented to the user support staff with several key pieces of information. Common facts that are often presented to or requested by the support staff are:

- System on which the application was running
- Userid that the suspect application was running under
- Nature of problem (incorrect results, application failure)
- Approximate time that the problem was noticed
- Exact error message, if applicable

For current single system applications, the problems are often submitted in real time. The user submits the program to a batch job or runs it interactively and the application fails or begins to behave improperly. The user is able to contact the support staff, who have the ability to log into the system and analyze its current health status.

A fundamental role of user services in a Grid environment will be to identify the location and general cause of failure; hardware, system software or user application. An application that utilizes multiple Grid resources will also entail which application and on which Grid resource it failed. If a job failure is found to be the result of a hardware or system software failure, the problem ticket can be directed to the appropriate support staff.

If job failure is found to be the result of the user application, 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. Isolating in the source code the application failure 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 discovering the application that failed, on which Grid resource it was running and if the failure was owing to the application or the environment. Discussing individual applications is a common step for any support staff.

3.4 User Service Level Agreements

User Service Level agreements are also an important factor for Grid applications. While not unique to Grid computing, their use in these 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.

3.5 Virtual Organization Support Agreements

In addition to the User Service Level agreement, there may be a need for system use agreements between organizations in the virtual organization. All centers monitor system usage to either charge groups who use their resources or to justify proper allocation of resources to a parent funding organization. Support staff will require some allocation of resources to carry out support tasks. The level of resources required will likely be non-trivial.

An agreement should be established between organizations allowing for mutual access to compute resources. It is anticipated that the majority of agreements will allow for mutual access at no charge. Due to the diverse applications of Grid system, however, this cannot be assumed, and an agreement on usage behavior should be drafted. Support services are a vital requirement for any production-computing environment and need to have adequate resources allocated.

3.6 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 resource 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
- Need to go back and investigate a failed job to identify the system where failure occurred

These are the general catalysts for developing the utilities outlined in this document. Consulting issues are covered in more depth in chapter 7 of the Grid User Services Common Practices document.

4 Grid Services Requirements

A set of information is important for users to know about in order to target the resources they wish to use. Frequently, users can use various resources to accomplish the task 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 system 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 knowledge enables the staff to assist 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 user's application. With possibly large quantities of resources allocated to a single job, this information needs to be collected and assembled automatically by user support tools.

Information about resources for some existing Grid configurations 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.

In addition to published information, diagnostic data is available though Grid service and system service log files. Because there are multiple instances of these services, multiple resource managers, for example, its log files must be kept in a known or published location. This availability will enable support utilities to find the information and parse it to extract details pertaining to a given Grid application. Consultants must also be able to view log information and standard output/standard error streams by some mechanism.

4.1 Resource Manager Logs

In Grid environments a service will be required on any computational resource to manage the execution of application processes and enforce the use policy established by the local organization. This service, which will be referred to as the resource manager, is now a batch or scheduler system, and users interact with the service via command line functions or a graphical interface. This interaction between user and scheduler may be abstracted out by resource brokers or portals, but the resource manager will continue to monitor and control job execution on local hardware, even if invisible to the user.

Because the resource manager is 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 or behavior of an application. A log reporting service or an API for remote access is desirable as a component in the jobmanager software. Applications can then access and query this information as needed.

In the absence of this functionality, service log files local to the system running the service can be examined for information. 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 can 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 that extra processing and I/O resources needed to run the repetitive searches could be significant.

The recommended 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 could be set up to publish the following information:

```
Mds-Software-deployment=jobmanager-[fork|pbs|lsf|condor]
        ObjectClass = ServerLocation
        ObjectClass = ServerLog
        ObjectClass = SchedLocation
        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 or Super User Do (SUDO) utility, described in section 5.5
- Group access permissions for a local Grid consultant account, 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 jobid, userid 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 resource 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 recorded only at the verbose logging level.

4.2 Scheduler Service Information

Current single resource scheduling technology covers a broad range of capabilities from simple first-in, first-out (FIFO) policies to advanced scheduling policies, which optimize resource utilization and minimize job wait times. 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 information will be useful for both users and support staff personnel.

A partial list of issues that might need to be diagnosed 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. Ability to access the logs will help identify possible problems with the software and postmortem analysis of failed jobs.

4.3 Implementation of Grid Monitoring Architecture

It will be discussed in section 5.3 that a tool should be developed, which, given a list of resources manually or automatically by a grid service job id or range of ids, fetches all of the resources that have been allocated and builds a text or graphical display showing status of every resource as well as interconnect performance.

Fundamental to this application is having the status and performance data needed to analyze system functionality and performance issues. The Performance Working Group of the Global Grid Forum is working coordinating the efforts of multiple groups to establish a Grid Monitoring Architecture (GMA).

The Global Grid Forum document “A Grid Monitoring Architecture”[2], and a sample implementation described in “A Simple Case Study of a Grid Performance System” [3] outline the structure of the monitoring tool.

The producing service should, at a minimum, record and publish in the directory service the following event types for given resources:

- CPU load
- Uptime
- Available memory
- Memory in use

Network parameters that record the interconnect between Grid resources should include the following event types:

- Theoretical peak network bandwidth
- Current network bandwidth

The monitoring tool can then collect, display and analyze the published data.

4.4 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.

Because future Grids comprise large amounts of hardware resources and span multiple physical organizations, these documents need to be 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[4] outlines some non-performance information that should be published. 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

4.5 Software Installation

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

5 Tools

Several tools would be 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

A fundamental problem of working in a distributed grid environment is that the software environment between multiple resources may not be configured properly to support a user's application. Possible issues encompass different software versions, OS versions, file system configurations, security settings, firewall or network configurations and local user account privileges.

A modular test suite is needed. Small, stand-alone applications should be available to verify each Grid service between selectable Grid resources. Critical also is an interface that allows support staff, and eventually users, to build a customized test suite of Grid services. The complete test suite need not be run to diagnose a possible problem. In the event of a failed application, only those services utilized by the suspect application need be selected for testing. Obviously, a user application will not function if there is a problem with the underlying Grid services.

The modular design also permits the entire Grid community to develop and submit the individual Grid service tests. Once a single unifying framework is established, individual tests can be designed as a module, which can be included or imported. Once imported, a menu-based system can be run; it will display all available tests, allowing the support individual or user to select which services to test. The framework can then execute, in sequence, all of the service tests and generate a text, xml or html report.

The master framework should be developed in a portable scripting language. Recommendations at the time this document was written are Python or Perl. Each test routine should be coded as an importable or loadable module. The framework will prompt the user to enter several pieces of information that will be made available to each test routine. The base set of parameters is

Local machine	(provide by script)
Remote machine	(required)
Remote file server	(optional)
Meta scheduler	(optional)

For the individual tests, the applications should verify the behavior of a given service. The script module should be self-sufficient and generate any files or data structures needed to test the service, such that including the test in the parent framework would be sufficient to complete the test. Logic would need to be included in each test to check if the optional parameters given above have been set or not.

5.2 Service Log File Collector/Browser

As discussed in section 3.3, a significant function of Grid application support will be postmortem analysis to determine the location and cause of an application failure. The majority of Grid applications, owing 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.

Therefore, it is necessary, as outlined in section 4.1, that diagnostic log data be retained on allocatable systems. All pertinent log files from all Grid resources will need to be analyzed. As recommended in section 4.1, verbose logging must be enabled. These logs, which might span multiple days, could be too large to browse manually. Extracting entries keyed to a specific user name or job will provide a concise summary, but general log entries that might be important would be missed.

One necessary feature is a choice of fetching the log files keyed by user name or jobid, or within a certain time window. Additionally, the application should have logic 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 resource 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 jobid should not be required, however, so a user support staff member should be able 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, one can locate log files 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 is 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 Grid Resource Browser

The primary function of this tool is to provide a snapshot of a group of Grid resources and their connections. This utility should be developed such that resources can be manually selected or automatically assembled, given a Grid jobid or range of ids. It will then fetch all of the resources that have been allocated and build a graphical display showing the status of every resource and interconnect. This application can then call other utilities to collect performance and load statistic data.

Some dynamic information can be measured by only by running an instrumented 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.

5.4 Standard Output and Error Monitor

One class of support problem is the status of a running application. For this type of problem, accessing the application’s standard output and standard error is important. With today’s applications, it is an important diagnosis tool for users to determine if a solution is progressing properly.

Resource managers should provide Grid-enabled access to standard output and standard error streams. By directing the streams to a Grid file system, Grid utilities will have access to output during a run. Support staff should also have the ability to authenticate and view the I/O stream.

For post-mortem analysis, it is desirable to have the additional functionality to set a residency time for the error and output streams in the Grid environment. This will permit support staff doing a post-mortem analysis to access the output streams for some time window after the run has terminated and not need to request the files from the user.

5.5 Grid-Enabled SUDO (Super User DO)

A tool 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.

5.6 Grid Debugger

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

A client-based structure should allow 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?

5.7 Information Browser

Multiple directory service browsers are also 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 subset 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 can trigger the directory browser to launch a web browser to display access and use policies, as discussed in section 4.4.

6 Capabilities

Support staff personnel will need a certain level of access to system logs and status information. This access 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.

A consultant may not need to enable or become a user. Most Grid applications will be complex enough that “becoming” the user to run the application will not be a practical diagnosis tool.

In situations in which it is needed, data conferencing technology can be used to share desktops. This method 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.

A separate document under development by the Grid User Services Research Group, titled “Trouble Ticket Exchange Specification“, is examining the requirements of a trouble ticket interchange standard. Support groups that support different communities of grid users, are assumed to all use some form of trouble ticket system to track their response to users’ problems. The object of creating a trouble ticket interchange format is to enable these groups to easily communicate and track problems, which span grids.

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 precludes 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 personnel 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 that 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 configuration system 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

10 Full Copyright Notice

11 References

[1] J. Towns, J. Ferguson, D. Frederick, G. Myers, Grid User Services Common Practices, published February 2003 as GFD-R.10. http://www.ggf.org/documents/GWD-I-E/GFD-I.010.pdf

[2] B. Tierney, R. Aydt, D. Gunter, W. Smith, M. Swany, V. Taylor, R. Wolski, A Grid Monitoring Architecture, published January 2002 as GFD-R.7. http://www.ggf.org/documents/GFD/GFD-I.7.pdf

[3] R. Aydt, D. Gunter, W. Smith, M. Swany, B. Tierney, V. Taylor, A Simple Case Study of a Grid Performance System, published May 2002 as GFD-R.8. http://www.ggf.org/documents/GFD/GFD-I-8.pdf

[4] G. Myers, Grid Constitution, published February 2003, AD Review. http://www.ggf.org/Public_Comment_Docs/Public_Comment_Documents.htm