

GStat 2.0: Grid Information System Status Monitoring

Laurence Field

CERN, Geneva, Switzerland

E-mail: Laurence.Field@cern.ch

Joanna Huang

ASGC, Taipei, Taiwan

E-mail: joanna@twgrid.org

Min Tsai

ASGC, Taipei, Taiwan

E-mail: min17502@gate.sinica.edu.tw

Abstract. Grid Information Systems are mission-critical components in today's production grid infrastructures. They enable users, applications and services to discover which services exist in the infrastructure and further information about the service structure and state. It is therefore important that the information system components themselves are functioning correctly and that the information content is reliable. Grid Status (GStat) is a tool that monitors the structural integrity of the EGEE information system, which is a hierarchical system built out of more than 260 site-level and approximately 70 global aggregation services. It also checks the information content and presents summary and history displays for Grid Operators and System Administrators. A major new version, GStat 2.0, aims to build on the production experience of GStat and provides additional functionality, which enables it to be extended and combined with other tools. This paper describes the new architecture used for GStat 2.0 and how it can be used at all levels to help provide a reliable information system.

1. Introduction

The Grid Information System is a mission-critical component of the EGEE [9] production infrastructure. It is a hierarchical system built out of more than 260 site-level and 70 global aggregation services. Users, applications and services use the system to discover which services exist and further information about their structure and state. To reliably fulfill these goals, the information system components must be functioning correctly and the information content should be correct.

Grid Status (GStat) [10] is the current monitoring component for the EGEE information system. It presents different overviews of the content found in the information system and performs various sanity checks on it. The results are used to identify potential problems with the information system and the Grid itself. The problems found can be brought to the attention of Grid Operators and System Administrators via an alert mechanism.

GStat evolved from a simple display showing the current status of the Grid into a more complex information monitoring and validation tool. Due to the increasing size and complexity of the EGEE infrastructure, it is necessary to re-evaluate the design of GStat. In addition GStat was designed as a centralized operations tool, and due to the European Grid Initiative [1] proposing a more decentralized model, it must be ensured that GStat can continue to operate in such an environment.

Section 2 of this paper outlines the general approach taken by Gstat with respect to information system monitoring and content validation. An alternative architecture for GStat 2.0 is presented in section 3 and the resulting implementation and initial feedback from deployment is discussed in section 4.

2. An Approach to Information System Monitoring and Validation

The initial approach used by GStat to validate the information system was to visualize the information content. The site-level aggregation points contain information representing a snapshot of the existence and state of the services running at that site. By visualizing this information from all sites, it is possible to identify problems manually through observation. This approach is also useful during troubleshooting whereby information can easily be found relating to an issue under investigation.

Experience gained through observation and troubleshooting can be used to identify common issues. In addition to increased understanding of the information model, automated checks can be defined. The primary check is validating that the information conforms to the information model. Although this is mainly covered by the information system components themselves, the tolerance level of the components may be higher than is required by certain use cases. For example, the component may check that a value is a string; however, the use case may dictate that the string has a specific format. Another example is that in the information model attributes may be defined as mandatory or optional. The supported use cases may require attributes to be present and such additional requirements on the information model will also require validation.

A common issue is with missing information and the ambiguity between entities disappearing and not existing. The two approaches that can be used to overcome this ambiguity is to either record the entities that have been seen in the information or to compare the entities in the information system with an external source. Both approaches will identify situations which would require further investigation, however they both have drawbacks. Recording entities will miss the condition where there is an initial problem with information being published into the information system. In addition when a service is decommissioned, this approach would falsely generate an error. Comparison with another source will depend on the accuracy of that source, however each additional source will improve the confidence in that value. Any errors generated will require further investigation to identify the reason for the inconsistency and hence the exact location of the error.

Even when it can be confirmed that the information is in the correct format and should exist, is the information reported valid? Although it is difficult to prove that a value is correct, there are four types of checks that can be employed to improve the confidence in a particular value. The first type of check that can be carried out is a logical test on the value. An example would be to check that the total number of cores in a computing cluster is negative or too large. The second type of check is a self consistency check on the value, for example, is the amount of free storage space larger than the total available storage space. The third type of check is a comparison with an external source. Finally, as issues are discovered and solved, a resulting check should be produced to detect if that situation occurs again in the future.

When contacting the site-level aggregation points, metrics such as the number of entries found and the time taken can be gathered. These metrics can be gathered periodically and their change monitored over time. This approach can give an indication of the health of the

component.

In summary, the checks described above can be categorized into Visualization, Content Validation and Infrastructure Monitoring. Using these checks, any anomalies can be reported to the Grid operations team by means of an alarm mechanism. These alarms can be integrated in to the day-to-day procedures used to operate the infrastructure.

3. GStat 2.0 Architecture

The GStat architecture, shown in Figure 3, is split into four main areas: Core, Validation, Monitoring and Visualization. At the center of the system is a database where the data model used defines the interface between the areas. GStat Core is responsible for extracting information from the information system and maintaining an entity cache. GStat Visualization uses this information for the structure of displays, and GStat Validation conducts various validation tests on the information content. GStat Monitoring uses the entity cache to configure a monitoring framework. GStat Visualization uses all the resulting information to present various views based on the different use cases. The separation of the different areas enables only the required parts to be used. For example for validating the information content, only GStat Core and GStat Validation would be required.

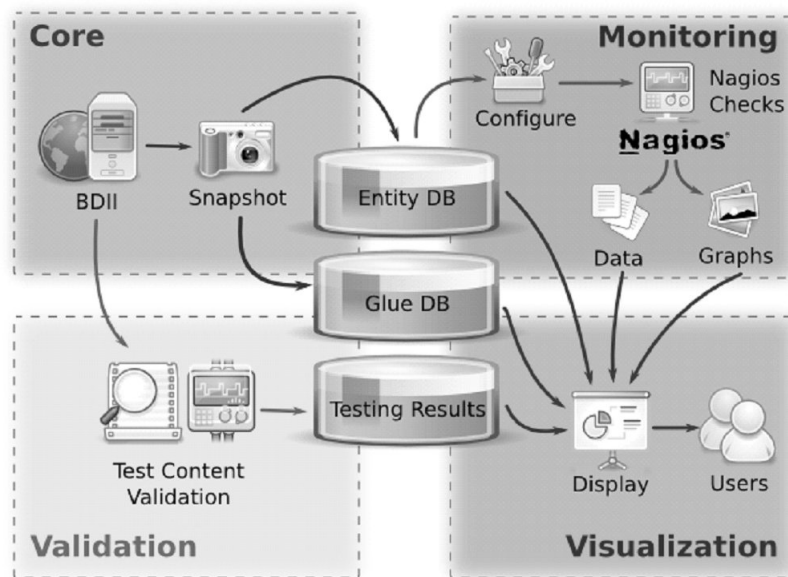


Figure 1. GStat 2.0 Architecture

3.1. GStat Core

GStat Core's primary functions are to initialize the database, maintain snapshot of the information system and update the entity cache. A snapshot of the information system is stored in the database, which can be used to find metadata about the sites and services in the infrastructure. This snapshot is created by running the snapshot script which also updates the entity cache. As the entity cache is created from the information system, the monitoring scope of the instance is defined by the information aggregation point that is queried. This enables multiple instances to be deployed, which monitor the system at various levels. On the one hand an instance could be configured to query a top-level aggregation point, however, an instance

could also be configured to only query one region if an aggregation point existed that contained information from that region. This approach enables GStat to be used in a decentralized manner where each region is able to manage its own instance. GStat Core also contains the common libraries and scripts that are shared amongst many other components.

3.2. GStat Validation

The GStat Validation contains a set of scripts that are used to validate an information source. These scripts use a common library to ensure that they all have the same "look and feel" and behave in a consistent manner. The source for the tests can either be an LDAP server or an LDIF file. The results of testing can be directed to different output channels; the main two being standard out and the GStat database. This mechanism has been designed so that it can also be executed on the command line and as such can be used directly by System Administrators and automated testing suites.

3.3. GStat Monitoring

GStat Monitoring ensures the integrity of the information system by monitoring each component in the information system. As each component is published in the information system, the components which need to be monitored can be found by querying the entity cache. This information can be used to configure a monitoring framework to configure monitoring probes which monitor the individual information system components.

3.4. GStat Visualization

GStat Visualization is a framework which can be used visualize the resulting information. Its aim is to simplify various tasks by providing common functions for obtaining the data and provide a common "look and feel" for the resulting web application. The visualizations fall into four main categories; the information in the information system, the information system components, experimental displays and error resulting from validation. Together they allow a user to browse the topology of the infrastructure and locate possible problems. The experimental displays provide different ways to present the information which are useful for dissemination activities. This goal is achieved by providing a common visualization library along with a number of specific visualizations.

4. Initial Deployment Results

The main function of GStat Core package is provided by the snapshot script. This script queries an information aggregation point and stores the result in the GStat database. Aggregation points in the EGEE infrastructure are implemented using the Berkely Database Information Index (BDII) [8], which is based on LDAP [11] and the information is described by the GLUE [6] schema. Therefore the snapshot script is required to translate the GLUE schema from an LDAP data model into a relational data model. Errors during this translation due to problems with the implementation of the GLUE schema in the EGEE information system are recorded in the GStat database. In addition, the script extracts the main entities from this model and updates the entity cache. GStat Core can be deployed in a stand alone mode that can be used for the purpose of maintaining a relational snapshot of the information in the information system and resulting topology of the infrastructure.

A common validation library was created for the validation scripts. The validation library contains common functions for obtaining information from LDAP sources and managing the testing results. Various validation scripts have been written to validate the information from these LDAP sources. Many of the scripts check for compliance with the GLUE schema and conformance to the common use cases found in the EGEE infrastructure. It is envisaged that

these scripts will be used in many different ways. System Administrators can use them to check their own installations, they can provide additional testing for software development processes and be integrated into other infrastructure monitoring tools.

GStat Monitoring leverages the Nagios [4] monitoring framework and Nagios probes have been created to monitor the information system components. These probes are common to GStat and the Nagios-based Multi-Level Monitoring Framework from the Operations Automation Team [7]. This not only reduces the development and maintenance effort for both projects but also enables the information system monitoring probes to be easily integrated into any Nagios-based deployments. The size and momentum of the Nagios community can be leveraged to provide additional tools for GStat such as graphing and alarms.

The visualization library prototype was developed around the Yahoo User Interface library [5], however there are on going investigations into the use of ExtJS [2]. The visualization library provides a common header and footer for the GStat 2.0 web applications. Functions such as dynamic tables are also included in the library which simplifies the use of such tables. A function for accessing data from both the GStat database and Nagios is provided by GStat Core. These simplify the mechanism for obtaining data for the web applications. Initial applications include an information system content browser, an infrastructure browser and an information system browser. Each enables the user to drill down on information and highlights possible errors with the information system and its content. A Google Earth [3] application was created to demonstrate features of the GStat 2.0 framework. This application dynamically creates a embedded Google Earth view, which show the sites participating in the EGEE infrastructure.

5. Conclusions

Grid Status (GStat) has been successfully used as the monitoring component in the EGEE project for over four years. Due to the increasing size and complexity of the EGEE infrastructure, it was necessary to re-evaluate the design of GStat taking into consideration new requirements raised by EGI.

The approach used to validate the information system and its content is split into three categories; Visualization, Content Validation and Infrastructure Monitoring. Information Content Validation is further split into four categories; logical tests, self-consistency, external-constancy and regression tests.

The GStat 2.0 architecture has been defined with these concepts in mind. It is split into four main areas: Core, Validation, Monitoring and Visualization. This split enables components to be re-used and integrated into other tools and procedures. A single information system aggregation point is used for the bootstrapping process. This enables GStat to function in a decentralized mode. Different instances of GStat can be deployed to monitor sites, regions or different infrastructures by configuration alone.

The use of a stand-alone validation script enables them to be used in many different ways. Leveraging the Nagios framework has not only simplified the development and enabled the rich content of the Nagios community to be leveraged, it has also simplified the integration of GStat into the Multi-Level Monitoring Framework from the EGEE OAT. The visualization framework provides a foundation for building web applications tailored to specific use cases. Information from the information system is available for visualization and the results from the validation and monitoring are also at hand. Further investigation into improved methods of visualization are able to build upon this foundation.

The new architecture for GStat 2.0 has enabled it to address the requirement for decentralized operations tools and in addition, the separated areas provide building blocks for other operations tools. Together, they provide a powerful tool for information system monitoring.

6. References

- [1] European grid initiative. <http://web.eu-egi.eu/>, 2008.
- [2] Extjs. <http://extjs.com>, 2009.
- [3] Google earth. <http://earth.google.com>, 2009.
- [4] Nagios. <http://www.nagios.org>, 2009.
- [5] Yahoo user interface. <http://developer.yahoo.com/yui>, 2009.
- [6] S. Andreozzi, S. Burke, L. Field, and B. Konya. Glue schema specification version 1.3. <http://glueschema.forge.cnaf.infn.it/Spec/V13>, 2004.
- [7] J. Casey. Operations automation team. <https://espace.cern.ch/sa1-share/oat/default.aspx>, 2009.
- [8] L. Field and M. W. Schulz. Grid deployment experiences: The path to a production quality ldap based grid information system. In *Proceedings of the International Conference on Computing in High Energy and Nuclear Physics (CHEP 2004)*, 2004.
- [9] B. Jones and F. Gagliardi. Building an infrastructure for scientific grid computing: status and goals of the egee project. *Royal Society of London Transactions Series*, pages 1729–42, 2005.
- [10] M. Tsai. Gstat. <http://goc.grid.sinica.edu.tw/gstat>, 2004.
- [11] K. Zeilenga. Lightweight Directory Access Protocol version 2 (LDAPv2) to Historic Status. RFC 3494 (Informational), March 2003.