Global Grid Monitoring: The EGEE/WLCG Case

Alexandre Duarte
European Organization for
Nuclear Research
Geneva - Switzerland
Federal University of Campina
Grande
Campina Grande - Brazil
Alexandre.Duarte@cern.ch

Piotr Nyczyk
European Organization for
Nuclear Research
Geneva - Switzerland
Piotr.Nyczyk@cern.ch

Antonio Retico
European Organization for
Nuclear Research
Geneva - Switzerland
Antonio.Retico@cern.ch

Domenico Vicinanza
European Organization for
Nuclear Research
Geneva - Switzerland
University of Salerno
Salerno - Italy
Domenico.Vicinanza@cern.ch

ABSTRACT

Grids have the potential to revolutionize computing by providing ubiquitous, on demand access to computational services and resources. However, grid systems are extremely large, complex and prone to failures. In this paper we present a tool able to to check if a given grid service works as expected for a given user or set of users on the different resources available on a grid. Our solution deals with the grid services as single components that should produce an expected output to a pre-defined input, what is quite similar to unit testing. Our tool, called Service Availability Monitoring or SAM, is being currently used to monitor some of the largest (maybe the largest) production grids available today.

Categories and Subject Descriptors

D.2.5 [**Software Engineering**]: Testing and Debugging—*Distributed debugging, Diagnostics, Testing tools*

General Terms

Measurement, Reliability, Experimentation, Verification

Keywords

Service Availability Monitoring, Grid Monitoring, Software Testing, EGEE, WLCG

GMW'07, June 25, 2007, Monterey, California, USA.

1. INTRODUCTION

Grids have the potential to revolutionize computing by providing ubiquitous, on demand access to computational services and resources. They promise to allow on demand access and composition of computational resources provided by multiple independent sources. On the other hand, grid characteristics, such as high heterogeneity, complexity and distribution (traversing multiple administrative domains) create many new technical challenges, which need to be addressed.

Among these technical challenges, failure management is a key area that demands much progress. Even fault diagnosis, a basic step in any failure management strategy, needs to see great improvement if we are to realize the grid vision. Today, when a grid user or administrator sees a failure in their screen, they have a very hard time in pinpointing the root cause of the failure. It may be the user's own application that has a bug, it may be that the user has requested a certificate whose lifetime was too short, it may be a configuration problem in some site that was used by the application for the first time or even that a disk on a machine next door has crashed. It may be a very large number of things. To further complicate things, error messages may be misleading. A recent study shows that even specialized users, such as system administrators, can spend as much as 25% of their time following wrong paths suggested by unclear error messages[**?**].

In this work we want to check if a given grid service works as expected for a given user or set of users on the different resources available on a grid. To achieve this objective we developed a framework that uses acceptance-like tests to help diagnose failures on the grid. In this new framework we deal with the grid services as single components that should produce an expected output to a pre-defined input. The framework is called Service Availability Monitoring or SAM, and is being currently used to monitor some of the largest (maybe the largest) production grids available nowadays.

We start by presenting in Section **??** a brief introduction on the gLite Grid Middleware, which is being used by the grid monitored with SAM. In Section **??** we present our solution, with some historical background and architectural descriptions. Further, in Section **??** we present a case study of the use of our tool to monitor one of the largest (maybe the largest one) grid in production today. In Section **??** we discuss some related works and finally, Section **??** concludes the paper with our final remarks.

2. THE GLITE MIDDLEWARE

The gLite middleware [**?**][**?**] was born from the collaborative efforts of more than 80 people in 12 different academic and industrial research centres as part of the EGEE Project[**?**]. gLite provides a leading-edge, framework for building grid applications tapping into the power of distributed computing and storage resources across the Internet. Its first version, gLite 1.0, was released on April 4th 2005. The latest version before converging toward a common architecture with the LCG[**?**] middleware was 1.5, released in January 2006. In May 2006 gLite 3.0 was released, merging LCG2.7 and gLite 1.5. This release contained all the services from LCG 2.7 with the addition of several components from gLite1.5. Starting from this gLite 3.0, there were no longer separate releases of the two middleware stacks it is being now used by EGEE and by WLCG (the worldwide LCG extension).

2.1 gLite architecture

The gLite services can be thematically be grouped into 5 service groups: Access Services, Security Services, Information and Monitoring Services, Data Services and Job Management Services. Among the gLite services one can distinguish user, site, virtual organization (VO), and global (i.e. multi-VO) scope where combinations are possible (authorization policies may for instance be enforced by the VO and the site). Although most services are managed by a VO, there is no requirement of having independent service instances per VO; for performance and scalability reasons service instances will in most cases serve multiple VOs.

2.2 Executive summary of the services

Some gLite services coming from the 1.5 release were included in the gLite 3. The LCG workload management components are also available in gLite 3 and all the services are now accessible from both tool sets in order to ensure a smooth upgrade from the LCG 2.7 and gLite 1.5 to gLite 3.0.

Access and Security Services:.

The prime aim of the Access and Security Services is identifying users, allowing or denying access to services, on the basis of some agreed policies. It provides a credential having a universal value that works for many purposes across several infrastructures, communities, VOs and projects. To carry out this task, gLite uses the Public Key Infrastructure (PKI) X.509 technology using Certification Authorities as trusted third parties.

Information Service (IS) and Monitoring:.

The IS provides information about the gLite resources and their status. The published information is used to locate resources and for monitoring and accounting purposes. Much of the data published to the IS conforms to a schema that defines a common conceptual data model to be used for resource monitoring and discovery. All the LDAP URLs used to query the information services running in each site are stored in a database called GOCDB.

Job Management System, Resource Broker, Computing Element and Worker Node:.

The Job Management Services collects information about the resource usage done by users or groups of users (VOs). The up-to-date information about the Services/Resources is gathered via sensors (Resource Metering, Metering Abstraction Layer, Usage Records). Records are collected by the Accounting System (Queries: Users, Groups, Resource). Within the services provided by the Job Management Service, the Computing Element (CE), represents some set of computing resources localized at a site (i.e. a cluster, a computing farm) that is responsible for job management: (submission, control, etc.) A CE provides a generic interface to the cluster, and the cluster itself, a collection of Worker Nodes (WN), the nodes where the jobs are run. One of the most relevant services among the ones provided by the Job

Management Services is the Workload Management System, a service running on a machine called Resource Broker. The RB is responsible for the distribution and management of jobs across sites. The purpose of the WMS is to accept user jobs, to assign them to the most appropriate CE, to record their status and retrieve their output. Jobs to be submitted are described using the Job Description Language (JDL), which specifies, for example, which executable to run and its parameters, files to be moved to and from the worker node, input files needed, and any requirements on the CE and the WN.

Storage Element:.

The SE is the gLite component which takes care of the Data Services, providing a storage back-end.

3. SERVICE AVAILABILITY MONITORING

SAM is a monitoring system that was developed based on more than two years of experience with providing high level monitoring tools for EGEE/WLCG grid infrastructure. The concept of high level monitoring emerged in EGEE/WLCG as the solution to manage the growing infrastructure that started with about 20 sites and quickly grew to 60, then more then 100 and ultimately beyond 200 computational sites. Number of sites and diversity of low level monitoring tools (a.k.a. fabric monitoring) made it impossible for a single operational body to know and understand the status of the whole grid and individual sites.

TestZone Tests.

The first approach to tackle this problem was a simple set of *bash* and *perl* scripts to perform a centralized testing of *Computing Elements* and the resources hidden behind, namely *Worker Nodes*. The system contained the following components: set of *bash* scripts with WN tests based on site certification script provided in LCG release notes; skeleton of the test job (JDL file and main script) that can be submitted to a site and execute all the tests; set of *bash* scripts to submit test jobs to all sites; collect outputs as text files, parse output files and generate reports and *perl* CGI to process output files and generate HTML reports of test results.

Site Functional Tests.

Site Functional Tests (SFT) was a direct successor of TestZone Test. The name was changed to better describe the functionality of the system and to avoid references to the concept of *Test Zone*, which was abandoned. The list of significant changes included: tests submission and results collecting code was rewritten from scratch; list of sites to test was being taken from GOCDB and top-level BDII; addition of several new tests; introduction of the concept of critical and non-critical tests and addition of a programmatically interface to site status information.

Site Functional Tests 2.

As a consequence of emerging grid operations a number of new monitoring tools were developed and popularized. One of the most important of them was GStat[**?**], a tool to monitor and analyse grid information system, namely BDII. To enable a common monitoring platform that would allow sharing test results and monitoring information between SFT and GStat, a new version of SFT with major architectural changes was introduced. SFT2 included the following new features: a universal relational data schema providing abstract representation of monitoring data, suitable for SFT, GStat and potentially other monitoring systems; single master test job script instead of several small test scripts; addition of a basic pre-execution check to validate the test environment and minimize false alarms before launching the tests against sites and publication of test results directly from WN by the job master script.

Service Availability Monitoring.

After EGEE/WLCG grid infrastructure had grown in terms of number of sites but also number of different service types, it become clear, that the model imposed by SFT that, despite its name, was only monitoring *Computing Elements* , was not giving enough information about the status of all important site services such as *Storage Elements*, *LCG File Catalogues*, etc. Because of the data model inherited from R-GMA the performance of SFT started to slowly degrade and the database soon became difficult to maintain. In addition there was intensive ongoing development of third-party monitoring frameworks for EGEE/WLCG that provided complementary information to SFT and/or covered services or areas not monitored directly by SFT. That is why a decision was made to extend SFT to a new system called Service Availability Monitoring (SAM) that would provide required features: *i*)optimized database schema for storing and processing test results, *ii*) concept of *sensors* as containers of tests targeted against different types of grid services, *iii*) concept of *standalone sensors* as third-party monitoring frameworks or test suites that could publish test results into SAM database in uniform format, *iv*) integration with other monitoring and operational tools and *v*) automatic service and site availability metrics calculation per VO based on critical tests selection.

3.1 Architecture

SAM is a system that although functionally replaces and extends SFT was redesigned almost completely from scratch based on the experience gathered with the previous tools. The architecture of the system, shown in figure **??**, displays its components logically divided into three independent layers: input, data storage and processing and output.

Input layer.

The input layer mostly consists of components responsible for executing regular tests against all grid services and delivering results. There are two possibilities: either tests are executed and results published by the default submission component or the equivalent functionality is provided by standalone monitoring tools that publish the results into SAM in a pre-defined way.

The framework provides a uniform platform for executing the tests and publishing test results to the central database. It has command line utilities to perform queries and to publishing test results to the underlying database through web services.

All the sensors in SAM are plug–in modules that communicate with the framework using fixed protocol. In the design of SAM we introduced two levels of hierarchy: sensors as containers and tests as individual code units (executables) which usually produce a single result record.

Storage and processing.

The components of SAM which are responsible for collecting monitoring data, storage and post-processing are installed on a central machine called *SAM Server.* The core of the system is the relational database[1] which holds all the information like: grid infrastructure description (sites, nodes, services, VOs and relations between them), test results, test criticality, availability metrics and application configuration.

Output.

The presentation layer of SAM contains a number of components that are accessing SAM database directly or indirectly (through XML data exports) and are even parts of external systems. The following are the three most important components. The *SAM Portal* is a reporting tool written in Python that displays individual test results by VO, service type, and region, as an HTML table with possibility of showing history of test results and the detailed log from test execution. The *GridView* visualization portal which shows configurable availability plots (intervals, VO-wise, site-wise, etc). The *COD Dashboard* is an external portal for Grid operations in EGEE/WLCG which is a front end to the alarm system and ticketing system.

4. CASE STUDY: MONITORING A GLOBAL GRID

A number of grid infrastructures are currently featured by SAM. As major examples we mention here those built within the EGEE, SEE-Grid [**?**], EELA[**?**], Health-e-Child[**?**], EuMedGrid[**?**], EuChina Grid[**?**], Baltic Grid[**?**] projects. In these contexts, which are in general different for scale, scope and purpose of the infrastructure, SAM platforms were deployed in slightly different configurations, according mainly to the number of sites monitored and to project's hardware and software resources.

On account of the number of its sites and Virtual Organizations, of its geographical spread, and of the complex structure of its operations, the EGEE/WLCG grid is by far the largest grid infrastructure among those featured by SAM services.

---

[1] In current implementation only Oracle DBMS is supported

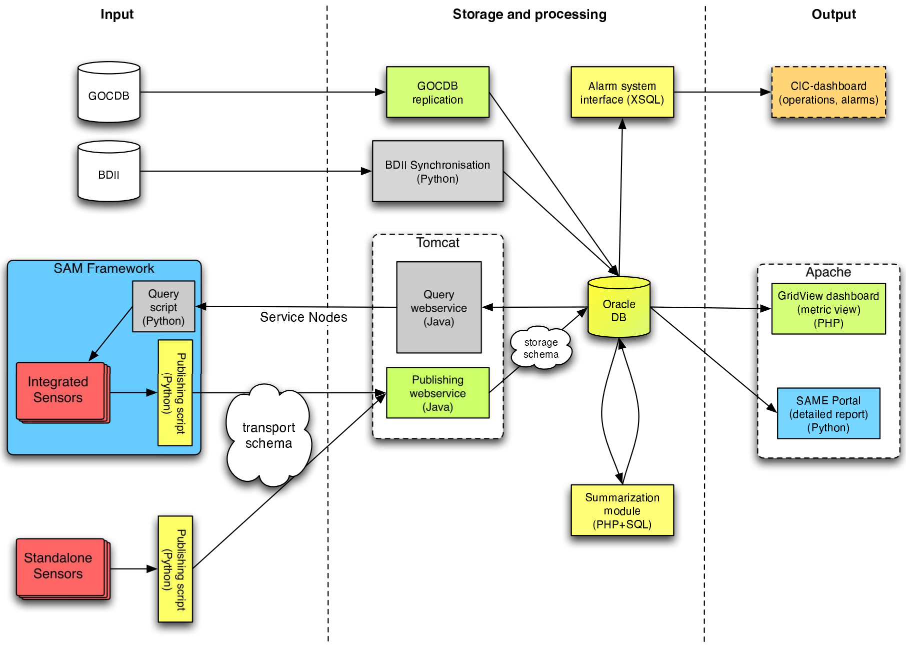

Figure 1: SAM Architecture

4.1 Grid operations in EGEE/WLCG

The EGEE/WLCG infrastructure is based on grid services provided by more than 200 sites distributed all over the world. The users of these services are organized in more than one-hundred different virtual organizations. In order to assure the overall quality of the service provided both in terms of availability and performance, the infrastructure was organized into 10 *federations* or *regions*. In order to coordinate the operations throughout the regions, an operational process was defined based on two main players: a network of Regional Operational Centres and the Grid Operators.

The Regional Operation Centre (ROC) holds the overall responsibility for the services run within its region. This means, in practical terms, to assure that all the sites in the region are operated in conformity to a set of agreed operation procedures. The Grid Operators (COD) are a distributed team in charge of providing an active and continuous monitoring of the availability and performance of the grid services. The key function of the COD is to detect issues affecting the grid services, to provide possibly a first analysis, to report existing problems to the relevant ROCs (generally via service tickets) and, finally, to validate the solution.

Site Certification.

In order to be integrated in the grid, a site has to demonstrate its technical suitability to run grid services at a convenient level of quality and, more important, not to introduce unexpected perturbations in the grid.

Just for example, it is technically possible to configure a grid site in such way that, independently on the actual amount and quality of its resources, it starts attracting user jobs in what is called a "black-hole" effect. In order to prevent this one well as other intentional or accidental security issues, each site that wants to join the grid has to undergo a preliminary testing and validation of it services, generally indicated as "site certification", to be done by the relevant ROC.

On account of SAM being among the first utilities available in the EGEE/WLCG context able to test the overall functionality of a whole grid site, SAM's test results are crucial in the certification procedures of most EGEE/WLCG ROCs. In the most common scenario, as one of the conditions for a site to be certified, the ROC wants a site to successfully pass the default SAM test across some days before allowing it to receive production jobs.

Site Monitoring.

Members of the COD have got a number of tools available to support their activity. Specifically in order to monitor the grid service availability, the main tool is the a dashboard that uses the SAM test results to raise alarms to be followed-up by the CODs with service tickets.

4.2 Monitoring activity by Virtual Organizations

In addition to the "institutional" monitoring activity done by the CODs, also some of the Virtual Organizations take active part in grid service availability monitoring by providing and maintaining specialized tests for VO-specific grid applications to be run, within the SAM framework, across a subset of supporting sites.

4.3 Measuring SAM effect

In terms of positive impact on the day-to-day grid operations, in the relatively short history of grid monitoring, and specifically dealing with the EGEE/WLCG context, we must say that the most significant improvement in the overall availability and stability of the EGEE/WLCG grid was undoubtedly reached before SAM went to operations, when the monitoring activity was featured by SAM's predecessor, SFT.

SAM took over SFT in production one year ago in order to better scale with the increasing number of monitored sites and tests to be run. In order to prove that SAM is indeed scaling quite well, we analysed the test results concerning the last year of operation, starting from the date when SAM went in production.

Figure **??** shows two plots. The upper one is the number of sites registered to the EGEE/WLCG project; the lower one is an indicator of the overall service availability. Specifically the overall service availability is calculated as the sum of the daily availability metric of each sites in production. The daily availability metric for a site is the percentage of time in the day during which the site has been fully operational. We tried to reduce the negative bias on the sum due to test results coming from still uncertified sites. Results of tests run across uncertified sites, in fact, are mixed-up in the database with those of certified sites. Since an historical record of status transitions for a site in not currently available, in order to correct this effect we decided to count in the sum the contribution of a given site starting from the first day in which the daily availability for that site resulted to be equal to 1. The consideration behind this assumption is that, when a site is available continuously for 1 day it is reasonable to foresee that its initial set-up phase is over and the site is getting close to the certification . In this way, without making a neat distinction between certified and uncertified sites, we meant to measure the "SAM effect" in terms of usefulness of the tool to the "operations" in broad sense, which means to the certification activity done by the ROC, as well as to the monitoring activity done by the COD. As an indirect proof of the robustness and scalability of the monitoring tool, Figure **??** demonstrates how the increasing number of monitored sites (about 50% more than the precedent year) did not affect the overall availability of the grid services although the manpower and effort spent for the monitoring activity was basically left unchanged.

5. RELATED WORK

Since we could not find any other tool able to run functional tests of grid services on a production environment we will present in this section some widely used tools to monitor grid infrastructures and point were we think that our solution is superior to them.

5.1 MapCenter

MapCenter [**?**], developed as part of the EU DataGrid project, is a monitoring application which provides web users a visualization of the availability and distribution of services throughout a Grid. It is intended as a grid administration tool for tracking availability problems.

MapCenter builds and periodically updates a model of the network services available in a grid, and provides this information in several logical views (sites, Virtual Organizations (VOs), applications, geographical) through a web interface. It is important to note that the information provided by MapCenter is about the availability of grid nodes and their services (e.g., the daemons of Globus Monitoring and Discovery Service (MDS), etc.); hence MapCenter does not keep details concerning configuration and utilization of resources. However, it does allow users to dynamically query an MDS server (using a PHP-based LDAP client), ping and otherwise interact with hosts (using CGI scripts). So, using MapCenter a given user can discover what services are alive in the grid but it there is no way to know if the available resources are properly configured to accept jobs from an specific Virtual Organisation or if the interactions among the different services during the lifetime of a grid job are working as expected.

5.2 GridICE

GridICE [**?**] was developed as part of the DataTag project in order to facilitate grid administrators work. It provides status and utilization information at Virtual Organization, site and resource level, as well as basic statistics derived from historical traces and real-time alerts, all through a web front-end.

GridICE has a centralized architecture where a main server periodically queries a set of nodes to extract information about the status of grid and network services, and the utilization of resources. The main server is based on Nagios, an open source, host and network service monitor that can be easily extended by the use of custom monitoring and notification plugins. GridICE has an MDS plugin for periodically querying Globus index information servers and information providers, whereas other plugins can be built, say, for RGMA. The collected information is stored in a DBMS and used to build aggregate statistics (e.g., total memory per site), trigger alerts and dynamically configure Nagios to monitor any newly discovered resources. End-users access the service through a PHP-based web front-end which includes logical views at VO, site and entity level as well as a geographical map.

As the reader may have already noticed GridICE resembles more a fabric monitoring tool then a grid monitoring tool. It indeed provides a grid view of the monitoring data but it is target to site administrators that are, usually, more interested in finding and solving problems with their resource. To a grid user point of view GridICE doesn't provide and kind of feedback about the suitability of a given resource to jobs of a given Virtual Organization nor any clue about the status of the interaction among different grid services.

5.3 Globus MDS

The Monitoring and Discovery Service [**?**], formerly known as the Metacomputing Directory Service, constitutes the information infrastructure of the Globus toolkit [**?**].

MDS 2.x is based on two core protocols: the Grid Information Protocol (GRIP) and the Grid Registration Protocol (GRRP). The former allows query/response interactions and search operations. GRIP is complemented by GRRP, which is for maintaining soft-state registrations between MDS components.

The Lightweight Directory Access Protocol (LDAP)[**?**] is adopted as a data model and representation (i.e., hierarchical and LDIF respectivelyLDAP Directory Interchange Format), a query language and a transport protocol for GRIP, and as a transport protocol for GRRP. Given the LDAP-based hierarchical data model, entities are represented as one or more LDAP objects defined as typed attribute-value pairs and organized in a hierarchical structure, called the Directory Information Tree (DIT).

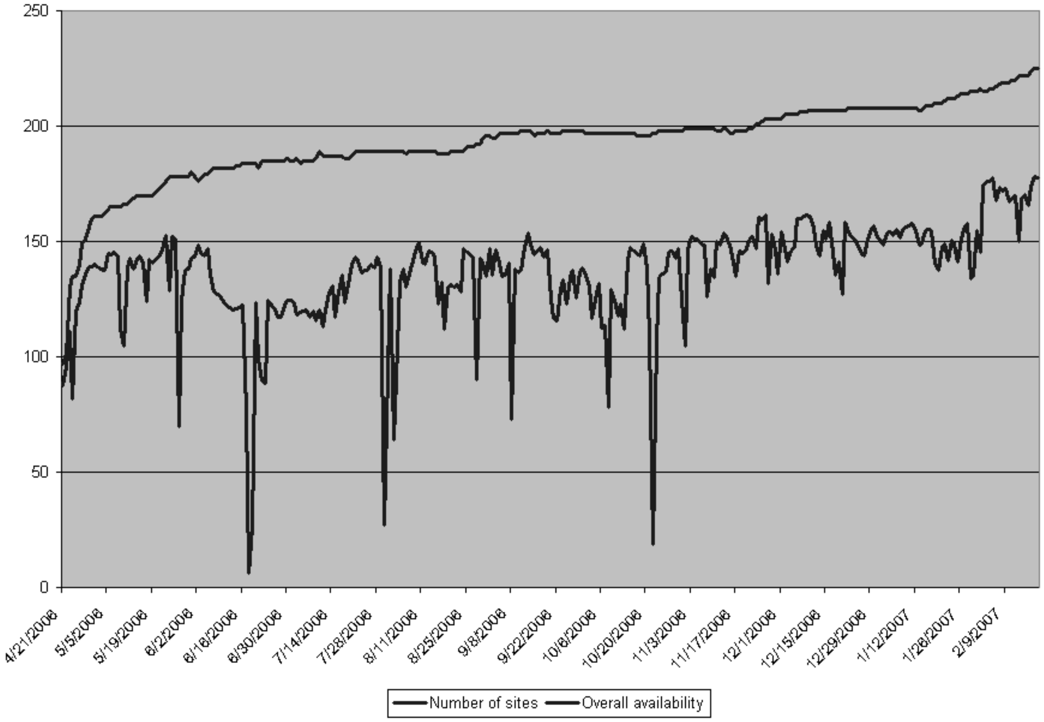

Figure 2: SAM availability

The MDS framework consists of information providers (sensors), Grid Resource Information Services (GRISproducers) and Grid Index Information Services (GIISrepublishers). Both producers and republishers are implemented as back ends for the open source OpenLDAP server implementation.

Producers collect events from information providers, either from a set of shell scripts or from loadable modules via an API. In addition, producers provide their events to republishers or to consumers using GRIP, and register themselves to one or more republishers using GRRP.

Republishers form a hierarchy in which each node typically aggregates the information provided by lower level republishers (and producers in case of first level republishers). Republishers use GRIP and GRRP as part of the consumer and producer interfaces, though custom implementations could offer alternative producer interfaces (i.e., relational). Several roles may be served by republishers, including the provision of special purpose views (e.g. application-specific), organization-level views and so on.

Consumers may submit queries to either producers or republishers, or discover producers through republishers, in any case using GRIP.

Again, this kind of grid monitoring solution, although being very flexible and extensible, can not be used to check the whole interaction chain of grid services during a grid job life cycle. Also, due to the fact that grid resources and under several different administrative domains, they have different management policies so it is difficult to be sure that the same configuration procedures were applied to each grid site and that all the resources in each site will act as expected for all virtual organizations that may have access to them.

5.4 GStat

GStat is a Grid Information System monitoring application [**?**] that is being actively used by the EGEE project. One GStat page exists for each EGEE Resource Centre and these pages are open to the public, i.e. no certificate-based access applies here. From the main GStat web page one can navigate to see the detailed pages that exist for every EGEE site.

The most interesting point here is the graphs provided by GStat, showing error (alert) levels and various other metrics, usually going as far back as the last 12 months. From these graphs one can examine the stability of a site, and possibly how long an error lasted.

A sites GIIS normally runs on the Grid Gate (the head node of the Computing Element) and collects information about all resources present at the site. GIIS entries are requested by the GStat server by running an LDAP search command every few minutes; the data returned to GStat is the reply from the GIIS of the corresponding Grid Gate.

The number of entries found varies from time to time due to the dynamic nature of the Grid (more specifically resulting from site configuration changes and changes of the software environment installed by the various VOs on the site). This means the number of normal entries can fluctuate and the sites information system could still be considered error-free and up to date; on some occasions however, the entries abruptly drop to zero (or quite lower than the current value), perhaps due to some network fault that causes time-outs or even disconnections, or a failure of the GIIS daemons running on the Grid Gate.

GStat can and is being successful used by EGEE to monitor its information systems but needs to be complemented with other tools since it does not monitor itself the status of the services or the interoperation of the different grid services.

6. CONCLUSIONS

We presented a framework that uses acceptance tests to help diagnose failures on the grid. In this new framework we deal with the grid services as single components that should produce an expected output to a pre-defined input. The framework, called Service Availability Monitoring or SAM, is being currently used to monitor some of the largest (maybe the largest) production grids available nowadays and proved to be helpful on improving the reliability of the monitored grid services.

7. REFERENCES

[1] S. Andreozzi, N. De Bortoli, S. Fantinel, A. Ghiselli, G.L. Rubini, G. Tortone, and M.C. Vistoli. Gridice: a monitoring service for grid systems. *Future Generation Computer Systems*, 21(4):559–571, 2005.

[2] BalticGrid. Develop and integrate the research and education computing and communication infrastructure in the Baltic states into the emerging European grid infrastructure. In *http://www.balticgrid.org*, 2007.

[3] R. Barrett, E. Haber, E. Kandogan, P.P. Maglio, M. Prabaker, and L.A. Takayama. Field studies of computer system administrators: Analysis of system management tools and practices. In *Proceedings of Computersupported Cooperative Tools*, pages 388–395. ACM Press, 2004.

[4] F. Bonnassieux, R. Harakaly, and P. Primet. Mapcenter: an open grid status visualization tool. In *Proceedings of the 15th International Conference on Parallel and Distributed Computing Systems*, 2002.

[5] S. Burke, S. Campana, A. D. Peris, F. Donno, P. M. Lorenzo, R. Santinelli, and A. Sciaba. glite 3.0 users guide. In *https://edms.cern.ch/file/722398/1.1/gLite-3-UserGuide.pdf*, 2006.

[6] Health e child. An integrated platform for European paediatrics based on a grid-enabled network of leading clinical centres. In *http://www.health-e-child.org*, 2007.

[7] EELA. E-infrastructure shared between Europe and Latin America. In *http://www.eu-eela.org*, 2007.

[8] EUChinaGrid. the euchinagrid initiative. In *http://www.euchinagrid.org*, 2007.

[9] EUMEDGrid. Empowering across the Mediterranean. In *http://www.eumedgrid.org*, 2007.

[10] S. Fitzgerald, I. Foster, C. Kesselman, G. von Laszewski, W. Smith, and S. Tuecke. A directory service for configuring high performance distributed computations. In *Proceedings of the Sixth IEEE Symposium on High Performance Distributed Computing*, pages 365–375. IEEE Computer Society Press, 1997.

[11] I. Foster and C. Kesselman. *The GRID - Blueprint for a New Computing Infrastructure*. Morgan Kaufmann Publishers, Inc., San Francisco, CA, 1999.

[12] F. Gagliardi, B. Jones, F. Grey, M.-E. Bgin, and M. Heikkurinen. Building an infrastructure for scientific grid computing: status and goals of the egee project. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 363(1833):1729–1742, 2005.

[13] The gLite Team. glite: Lightweight middleware for grid computing. In *http://cern.ch/glite*, 2007.

[14] M. Lamanna. The lhc computing grid project at cern. In *Proceedings of the IXth International Workshop on Advanced Computing and Analysis Techniques in Physics Research*, pages 1–6, November 2004.

[15] SEE-GRID. South eastern European grid-enabled e-infrastructre development. In *http://www.see-grid.org*, 2007.

[16] The GStat team. Grid statistics (gstat) description. In *http://goc.grid.sinica.edu.tw/gstat/filter_help.html*, 2006.

[17] M. Wahl, T. Howes, and S. Kille. Lightweight directory access protocol. In *RFC 2251 http://www.ietf.org/rfc/rfc2251.txt*, December 1997.