Evolution of SAM in an Enhanced Model for Monitoring WLCG Services

David Collados, John Shade, Steve Traylen (CERN), Emir Imamagic (SRCE)

CERN IT/GD. Geneva 23. CH-1211, Geneva, Switzerland

SRCE – University Computing Centre, Josipa Marohnića 5, 10000, Zagreb

David.Collados@cern.ch, John.Shade@cern.ch, Steve.Traylen@cern.ch, eimamagi@srce.hr

Abstract. It is four years now since the first prototypes of tools and tests started to monitor the Worldwide LHC Computing Grid (WLCG) services. One of these tools is the Service Availability Monitoring (SAM) framework, which superseded the SFT tool, and has become a keystone for the monthly WLCG availability and reliability computations. During this time, the grid has evolved into a robust, production-level infrastructure, in no small part thanks to the extensive monitoring infrastructure which includes testing, visualization and reporting. Experience gained with monitoring has led to emerging grid monitoring standards, and provided valuable input for the Operations Automation Strategy aimed at the regionalization of monitoring services. This change in scope, together with an ever-increasing number of services and infrastructures, make enhancements in the architecture of existing monitoring tools a necessity. This paper describes the present architecture of SAM, an enhanced and distributed model for monitoring WLCG services, and the required changes in SAM to adopt this new model inside the EGEE-III project.

1. Introduction

The origins of the Service Availability Monitoring (SAM) framework go back to 2005 and the EGEE[1] (Enabling Grids for E-Science) project. There was a need to understand, analyze and improve the computational infrastructure put in place by the LHC Computing Grid[2] (LHC) project for the extraction, storage, distribution and analysis of the data generated by the Large Hadron Collider[3] (LHC) experiment. The role of SAM was to monitor the functionality of the grid services to improve their availability. Right from the beginning, the use of SAM proved to be very beneficial, not only because any site or service manager could visualize the behaviour of their services, but also because LCG management could evaluate the performance of each of the grid sites. These advantages made more and more sites interested in the use of this tool, not only inside the EGEE project, but also within the American Open Science Grid[4] (OSG) community.

At the present time, SAM is being used by more than 330 certified and production sites in 48 countries to monitor more than 3,400 grid services. This global use of the tool, which was not initially designed to support different projects and infrastructures, has brought new challenges and problems that we have to be dealt with. Although SAM is used today in a number of grid infrastructures such as

EGEE, OSG and NDGF[5], and by several multinational grid projects like WLCG[2], Health-e-Child[6] and BalticGrid[7], there are different architectural constraints that must be amended to better support the current load of the system, whilst making it more flexible and scalable. At the same time, due to the evolution of the European grid infrastructure from EGEE to a multitude of national grid initiatives, the resources and responsibility for grid monitoring must move away from a centrally administered monitoring function. These are the main reasons why changes in the SAM architecture and in the grid monitoring strategy are necessary.

In this document, we present the existing SAM architecture used in production. At the same time, we expose the different reasons that make necessary an update of the current grid monitoring structure. Since more than one year now, we have developed a new grid monitoring strategy, lead by the Operations Automation Team[8] (OAT) and pushed by the SA1 activity inside EGEE. This paper describes the new monitoring strategy, why is it required, which are the existing deadlines for the different objectives, which are the objectives already developed and the pieces that still need to be finished in the coming months.

2. The SAM Monitoring Architecture

2.1. The existing monitoring framework

The current implementation of the SAM monitoring infrastructure follows a centralized architectural model. This had some advantages in the past, like for example having a controlled and consistent set of tests that were submitted regularly (every hour) to all services under the OPS (operations) VO credentials and to collect all the results from a central location to compute status and availability of the services. This model, as can be seen in figure 1, is composed of the following services:

- A **topology repository**: where we collect grid topology information (sites, service endpoints, service types, VOs, service-VO mappings, etc) from the authoritative information providers for EGEE and OSG (GOCDB[9], OIM[10]) and the BDII. We do this via data synchronizers that are run periodically.
- The SAM **database**: where we store the definition and the output of the metrics after execution.
- The SAM **web services**: used to extract information from the database and to publish new tests and tests' results into the database.
- A **tests' submission framework**: in charge of submitting the test to the different grid services in a scheduled way.

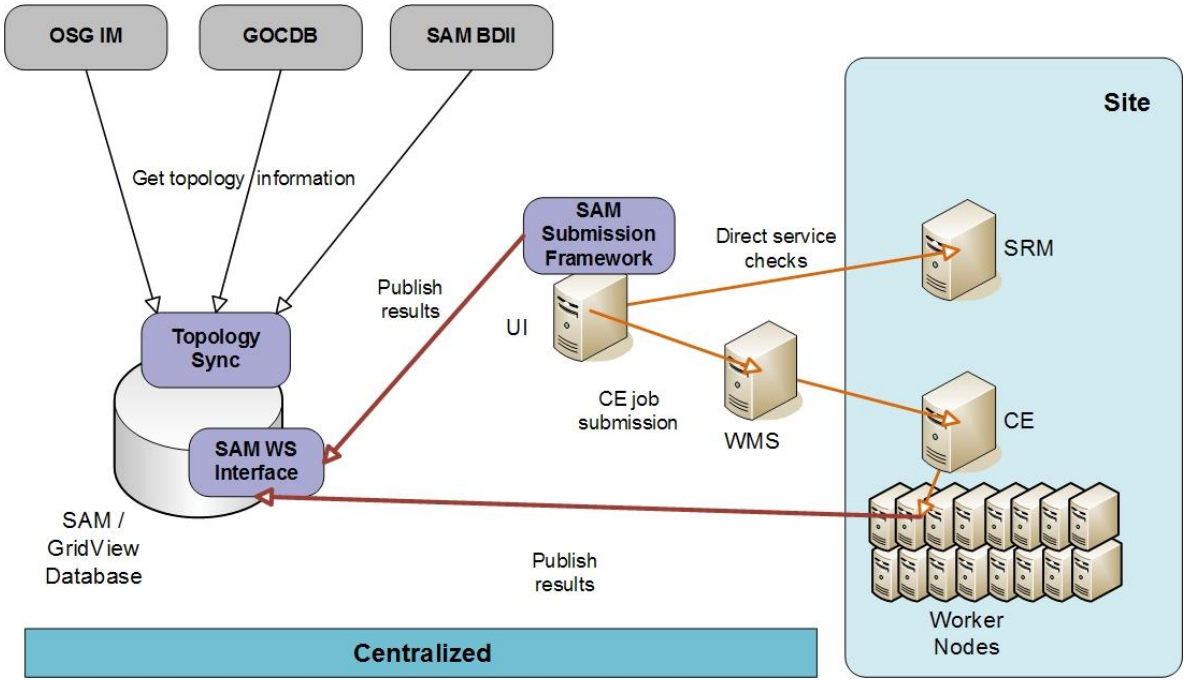

Figure 1: The Existing SAM Architecture

During the submission process, some of the tests are run directly from the UI (by contacting specific network ports on the target services). On the other hand, the Computing Element (CE) tests are submitted from the User Interface (UI) to the CEs via the Workload Management System (WMS). They land on Worker Nodes (WN), execute, and then contact the SAM Web Service (WS) to publish their results. There is no retry mechanism if the WS is unavailable. In parallel, the UI gathers the jobs' output from the WMS, and sends them to the publishing service.

2.2. Why do we need a new architecture?

After running SAM during the last four years, it has become clear that some parts of its architecture need to be redesigned. There are several reasons for this:

The SAM framework was originally designed to test services in one single grid infrastructure (EGEE). As we already mentioned, we are currently supporting **services** defined in several infrastructures, and some architectural changes are needed to make this support more reliable and flexible in view of a future increase in resources being monitored. An example of a necessary change is the database schema.

We need to improve the reliability of the grid and we can do this by giving site administrators better tools. Many sites have no fabric monitoring, and rely on the central SAM tests to see whether their grid services are functioning correctly. However, these run at a low frequency (once an hour), and the *Mean Time To Repair* a broken service would definitely be lowered if local tests, with local alerting, were carried out. For this, we want to present standardized and well-known open source tools to the site managers for local grid services monitoring.

At the moment, the SAM monitoring infrastructure is hosted centrally at CERN, and is developed and managed by staff funded by the EGEE project. During 2010, in a post-EGEE world made up of a collection of National Grid Initiatives, the number of entities that are responsible for their own

monitoring and availability statistics will increase significantly. Therefore, the current tools will have to be modified to fit a distributed model where infrastructures will be managed by NGIs rather than a central entity.

We have a central SAM infrastructure testing all grid services once an hour. This central infrastructure has advantages and disadvantages. One of the disadvantages is a reliance on the WAN, although the grid does that anyway, and we are monitoring the grid from a user's point of view. The network also places an inherent limit on the frequency at which tests can be run, since they all emanate from a central location.

Consistent set of tests under central control: By packaging the tests at each run, we can guarantee that all sites run the same versions of the tests at the same time. This may seem obvious, but it becomes a problem when considering what happens in a distributed environment with no central control.

One algorithm for calculating availability: This has the advantage of being simple and easily understood, but the complexity of the grid, with its wide variety of users with differing goals, means that the one-size-fits-all approach has its limitations. In particular, the HEP VOs have more stringent availability criteria that simply checking if the underlying infrastructure is operational.

OPS and VO-specific tests: SAM tests the availability of the infrastructure, but the VOs are more interested in the usability of the infrastructure (for example, ensuring that VO-specific software is in place). Therefore, we have tests that are supplied not only by the SAM team, but by VOs themselves, checking for instance, data distribution and data analysis. Moving to a new architecture needs to take this into account.

2.3. Why do we need a new architecture?

Apart from the technical reasons that we have just explained and that justify some changes in the existing monitoring framework, we have identified other limitations of the existing model, for which we have defined and started to implement solutions.

Elapsed time before site administrators are alerted to problems: As mentioned earlier, the hourly testing interval, along with the delays in the central alarming system mean that site administrators do not receive timely notifications of problems. If a problem occurs immediately after a test is run, it will be at least an hour before the site administrator is alerted. By running higher-frequency tests locally, with local alarms (e-mail, SMS), we can drastically improve the situation. Not only that, but a higher sampling frequency implies that availability statistics could be made more accurate.

Sites are blind if central monitoring fails: Sites that do not have local monitoring are entirely dependent on the central monitoring infrastructure for discovering problems. If this fails, sites will have to rely on user complaints to discover problems. The second thrust of our monitoring strategy is therefore to provide sites with a tool that combines both grid and fabric monitoring (faulty fans, cabinet temperature, disk space etc.).

Possible scaling issues if number of services increases significantly: Although we have not yet encountered significant scaling issues, they will appear as the grid grows. The current one hour sampling period is partly dictated by the central architecture and the large number of (sometimes sequential) tests that have to be launched.

During SAM outages, test results are lost: The lack of a retry mechanism in the SAM clients means that if the Web Service is unavailable when they try and publish their results, the results are lost. To

address this, we decided to use ActiveMQ messaging technology [11]. This effectively acts as a store and forward network transport, with guaranteed delivery. Messaging technology is described in the OAT Strategy [12] and is the building block that all operational tools will be using for inter-communication in the new architecture. Clients talk to brokers using a dedicated port, with the ability to use HTTP if necessary. This avoids many of the firewall-related problems when Worker Nodes attempt to send results back to the monitoring server.

No history of grid topology: Grid services come and go in the fluctuating grid infrastructure, and it is important that snapshots are taken so that availability calculations can be (re-)done using a consistent topology. The ability to do this is lacking in the current SAM topology repository. We are therefore developing a new component, called the Aggregated Topology Provider (ATP) [13]. As the name implies, it will gather topology information from a list of authoritative sources, and will provide a query interface that is able to return time-indexed topology information (i.e. which services were present in the grid at a specific time).

Non-flexible availability calculations: Different users have different concepts of what constitutes an available service or site. For example, ATLAS analysis jobs have different requirements from users of the Biomed VO. In order to allow multiple availability algorithms, we are introducing the Metric Description Database. This will contain all the different tests which can be combined in different ways in order to calculate a status. Each test description will include attributes such as version number, parameters, and pointers to documentation.

3. The Enhanced WLCG Monitoring Strategy

Figure 2 shows the proposed new monitoring architecture, based on a hierarchy of Nagios servers. The SAM database is split into three new components with additional functionality. This split is essential for enabling automatic Nagios configurations to be generated. NCG queries the ATP to discover which resources at a site (or in a region) should be monitored, and queries the MDD to discover how to monitor them (i.e. which tests to use). Test results are published to the Message Bus [11] – either to a queue for a specific consumer, or as a broadcast in cases where multiple entities would be interested in the data (e.g. a regional and project-wide database).

The middle component depicts the regional-level ROC Nagioses. As part of the evolutionary process, eleven such servers have been set up at CERN. They run Nagios tests which are equivalent to the SAM tests used for availability calculations (CE, SRMv2 and s-BDII), and the resulting data is read from the Message Bus into a separate instance of the current SAM database. GridView will then run the standard availability calculations, and we will be able to compare the results with those obtained from the current architecture. Demonstrating equivalence will be a key project milestone.

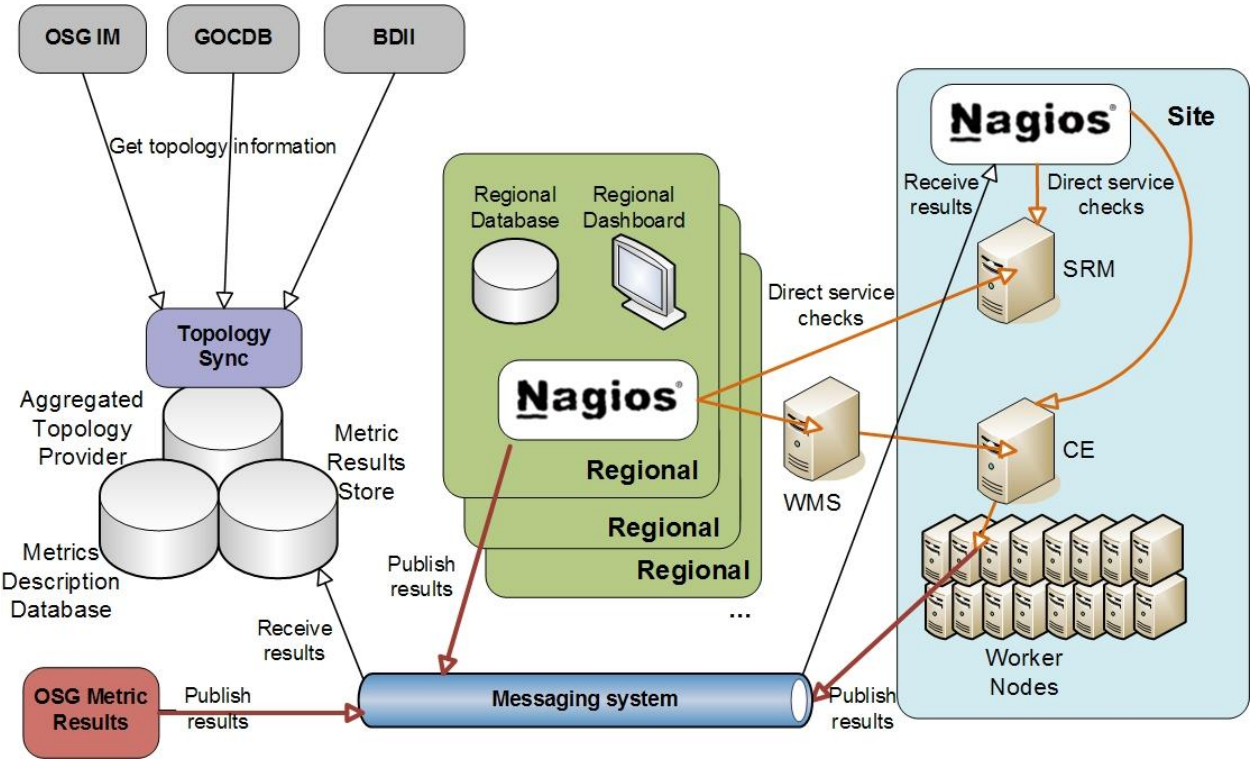

Figure 2. Enhanced Grid Monitoring Architecture

3.1. What is already in place?

We have adopted Nagios for site monitoring: There has been widespread acceptance of our choice of Nagios as the monitoring product to use. Several site administrators were already using it for fabric monitoring, and by providing a customized, YAIM-installable, site Nagios package, it is now in use at over 30 sites.

Thanks to the excellent work of the core OAT members, an automatic configuration generator is able to produce an installation that is tailored to the installing site. This hides the complexity of setting up and configuring Nagios from the site administrator, and has been instrumental in removing any initial reluctance to adopt a new product.

As soon as a site installs Nagios, it is able to receive the current SAM test results as passive checks. Site administrators are thus able to view all the monitoring information that is available for their site in one, easy-to-use interface.

Hundreds, if not thousands, of probes are available in the community. Although site administrators are free to choose any that suit their local needs, NCG removes the burden of selecting the appropriate ones for grid monitoring by configuring them automatically depending on the services that a site provides.

Work is already well-advanced in terms of converting existing SAM sensors into Nagios probes. A wrapper has been developed to do this automatically for simple sensors, and help is available to VOs who wish to convert their own SAM tests.

Regional Nagios package also developed: Pnp4nagios and NDOUtils have been added to the standard open source Nagios package. The former is a GUI for viewing graphs of the collected data, and the latter is a MySQL database which collects all Nagios events and test results. The regional Nagios is intended to be a devolved version of the central SAM framework. Each region will run a set of standard tests against its sites, and collect the data from which availability calculations can be done.

The current deployment scale can be seen in Table 1. It shows for instance that the EGEE ROCs are currently monitoring 171 sites with the WLCG monitoring strategy. The number is obtained by observing the number of sites for which metric results appear on the ActiveMQ message bus. It is envisaged that eventually project level monitoring will disappear and ROC and subsequently NGI monitoring will become the backbone of the monitoring infrastructure.

Nagios is deployed at the multiple layers of project, ROC or site within the EGEE infrastructure. Table 1 shows the number of sites that are being monitored by each of the layers.

Level of Naigos Monitoring	Number of Sites
Project Level Monitoring of Sites	279
EGEE ROC Level Monitoring of Sites	171
Site Level Monitoring of Sites	114

Table 1. Number of sites monitored by each monitoring layer.

We're using Active-MQ messaging technology: Several EGEE regions have already committed to running a message broker. This network of brokers will need to be run by grid operations as a reliable backbone. For the moment, CERN, INFN (Italy) and SRCE (Croatia) are providing the messaging infrastructure used by the operations tools.

A generic consume2oracle component has been developed to extract data from the Message Bus and insert it into Oracle and MySQL databases. It is being used to populate the test SAM database instance that is being run at CERN with Nagios test results.

GridView/SAM topology objects separated: This work was done as a prelude to developing the new ATP and MDD databases. Several overlapping tables were consolidated, and additional user accounts were created based on well-defined roles.

SAM Database exposed to Nagios for use by NCG (VO mappings, BDII and GOCDB): In order for Nagios to test the same services as SAM currently does, it needed to use GOCDB topology information, augmented with VO-specific information, and data from other infrastructures such as OSG. To this end, the contents of the SAM/Gridview topology database were made available to NCG through XML feeds.

Downtimes & user roles from GOCDB fed to Nagios instances: Nagios uses downtime information to suppress notifications. Since EGEE downtimes are declared in GOCDB, GOCDB's new programmatic interface was used by NCG to download downtime information to the local Nagios instance. Ultimately, we expect GOCDB to broadcast downtime information on the Message Bus but, in the meantime, this is a nice example of the integration of two operational tools.

3.2. What lies ahead?

ROC-level Nagios based monitoring available: By the end of April, we should be able to demonstrate equivalence in the availability calculations using SAM tests and Nagios tests.

The design work for the Metric Description Database and ATP has been completed, and prototype implementations exist. By July, we hope to have the integration with NCG completed, such that the submission framework fully uses the ATP.

Efforts are also underway to have a 'SAM Portal' level of visualization available for the regional Nagios instances and their associated Metric Results databases. We are collaborating with Open Science Grid to produce a customized version of their MyOSG portal for this purpose. Both parties are keen to work together on a coordinated approach to displaying test results, and the MyOSG portal uses state-of-the-art technology to easily integrate with Netvibes, i-Google and other popular frameworks.

Each regional Nagios will have its own Metric Results Store implemented in MySQL, but it is foreseen that the project-level database(s) will use Oracle. The project's metric store will be used by GridView for project-wide availability calculations. The goal is to keep the database schema identical for both cases. Nagios installations can also use the optional NDOUtils database and 3rd party visualization tools such as pnp4nagios and NagViz for additional views.

4. Conclusion

By selecting commodity software and interfaces which already have a large following in the community, we are providing a robust solution and limit the amount of support that has to be provided in-house. The original Service Availability Monitoring architecture has been realigned to encompass the organizational changes that will take place as we move towards EGI.

We provide a solution that scales from the site-level upwards. It provides the means for sites and ROCs to better monitor their grid services, ultimately contributing to a more available grid infrastructure.

References

[1] EGEE. Enabling Grids for E-sciencE. http://www.eu-egee.org, 2007.
[2] LCG. LHC Computing Grid. http://lcg.web.cern.ch/LCG/, 2009.
[3] LHC. The Large Hadron Collider. http://lhc.web.cern.ch/lhc/, 2009.
[4] OSG. Open Science Grid. A national, distributed computing grid for data-intensive research. http://www.opensciencegrid.org/, 2009.
[5] NDGF. The Nordic Datagrid Facility. http://www.ndgf.org/ndgfweb/home.html, 2009.
[6] Health-e-child. An integrated platform for European paediatrics based on a grid-enabled network of leading clinical centres. http://www.health-e-child.org, 2007.
[7] BalticGrid. Develop and integrate the research and education computing and communication infrastructure in the Baltic states into the emerging European grid infrastructure. http://www.balticgrid.org, 2007.
[8] OAT. The Operations Automation Team. https://espace.cern.ch/sa1-share/oat/default.aspx. 2009.
[9] GOCDB. The Grid Operations Centre Database. http://www.ukiroc.eu/content/view/87/250/, 2009.
[10] OIM. The OSG Information Management System. http://oim.grid.iu.edu/, 2009.
[11] Enterprise Integration Patterns: Designing, Building, and Deploying Messaging Solutions

(Addison-Wesley Signature Series), by Gregor Hohpe and Bobby Woolf.
[12] Operations Automation Strategy document: https://edms.cern.ch/document/927171
[13] J. Casey, D. Collados, R. Kalmady and others. Grid topology repository for WLCG Monitoring Infrastructure. International Conference on Computing in High Energy and Nuclear Physics (CHEP'09), Prague, Czech Republic, March 2009.