Journal of Physics: Conference Series **664** (2015) 062054 doi:10.1088/1742-6596/664/6/062054

WLCG Monitoring Consolidation and further evolution

P Saiz[1], A Aimar[1], J Andreeva[1], M Babik[1], L Cons[1], I Dzhunov[1], A Forti[2], A di Girolamo[1], E Karavakis[1], M Litmaath[1], N Magini[1], L Magnoni[1], H Martin de los Rios[1], S Roiser[1], A Sciaba[1], M Schulz[1], J Tarragon[1] and D Tuckett[1]

[1]CERN, European Organization for Nuclear Research, Switzerland

[2] University of Manchester, UK

Email: Pablo.Saiz@cern.ch

Abstract. The WLCG monitoring system solves a challenging task of keeping track of the LHC computing activities on the WLCG infrastructure, ensuring health and performance of the distributed services at more than 170 sites. The challenge consists of decreasing the effort needed to operate the monitoring service and to satisfy the constantly growing requirements for its scalability and performance. This contribution describes the recent consolidation work aimed to reduce the complexity of the system, and to ensure more effective operations, support and service management. This was done by unifying where possible the implementation of the monitoring components. The contribution also covers further steps like the evaluation of the new technologies for data storage, processing and visualization and migration to a new technology stack.

1. Introduction

The Worldwide LHC Computing Grid (WLCG) [1] is a global collaboration of more than 170 computing centres in 41 countries, linking up national and international grid infrastructures. Monitoring the status of the sites and services of this distributed infrastructure is a critical and non-trivial task. During the years, several monitoring solutions had been implemented to provide views of different areas. Different teams implemented these solutions, and they evolved in separate ways. The situation on June 2013 was that the effort required to support and maintain the WLCG monitoring solutions was bigger than the expected size of the team who would do it. Due to this, the 'WLCG Monitoring Consolidation Project' was created. The project had to perform a critical analysis of what was monitored, the technologies used and the deployment and support models, and to propose and implement a technical solution that would offer similar quality of service with a reduced effort.

The goals of the project were to:

1. Reduce the complexity of the system.
2. Ensure simplified and more effective operations, support and service management.
3. Encourage an efficient deployment strategy, with a common development process.
4. Unify, where possible, the implementation of the monitoring components.
5. Use, wherever possible, the same solutions as in the Agile Infrastructure Monitoring [4].

The main objective of the project was to get to the point where the effort required for WLCG monitoring can be reduced to half of its initial level.

The WLCG Monitoring Consolidation group was composed of experiments' representatives, operations representatives, a CERN Agile Infrastructure Monitoring representative, and the members of the IT-SDC-MI section, who are responsible for the development, maintenance and support of the WLCG Monitoring tools.

The project took into consideration the needs of all the stakeholders, in particular by:

- Reviewing all the existing monitoring applications supported by IT for the WLCG.
- Gathering requirements from experiments, regarding the remote testing of the distributed sites and services.
- Understanding the needs of the site administrators, answering the famous question 'I like to see at a single display how my site is working for all LHC VOs.
- Reviewing the test submission part, both for stress testing and for functional testing.

Once that was done, the project members:

1. Defined an overall architecture of the system.
2. Defined priorities for any required further development.
3. Agreed on a deployment and support model.
4. Used, where possible, a common framework and common implementations for all the components.
5. Provided a common web UI to WLCG monitoring applications.

The project was been divided in two phases. During the first phase (July to October 2013), the group performed an analysis of the initial status and agreed on a plan to follow during the second phase (October 2013 to December 2014).

2. Initial situation

The working group started by taking an in depth look at the current monitoring applications supported by the CERN IT-SDC group and their usage. The review identified a couple of applications that were not necessary to maintain anymore.

The next step, once the functionality needed was well defined and understood, was to identify the areas where applications could be combined, offering the same, or even better functionality, while reducing the support effort. In particular, the area of Site and Service Monitoring was identified as the one with the most potential for optimization. There were several applications (SAM [3], SSB [6] and SUM), which offered overlapping functionality, and it was agreed that the monitoring infrastructure would benefit if they could be integrated.

The initial schema of the monitoring applications on the Site and Services infrastructure can be seen in Figure 1.

There are three main applications used for the Site and Services monitoring:

- SAM (Service Availability Monitoring). This application was originally conceived as a distributed testing and reporting infrastructure. It sends periodically tests to services to evaluate their behavior. Based on the results of those test, SAM creates monthly site availability and reliability reports. These reports are propagated to the WLCG and EGI communities.
- SSB (Site Status Board). The SSB provides a framework where users can publish their data, and the SSB stores its evolution over time. It was created within the Experiment Dashboard Framework [2]. It was designed as a very flexible metric store, allowing the definition of new metrics or instances to be monitored in a very dynamic way.
- HammerCloud (HC) [5]. HC is a Distributed Analysis testing system. It provides the submission framework for the remote tests, repository for the test results and user interface. The system can be used for functional testing and for stress testing.

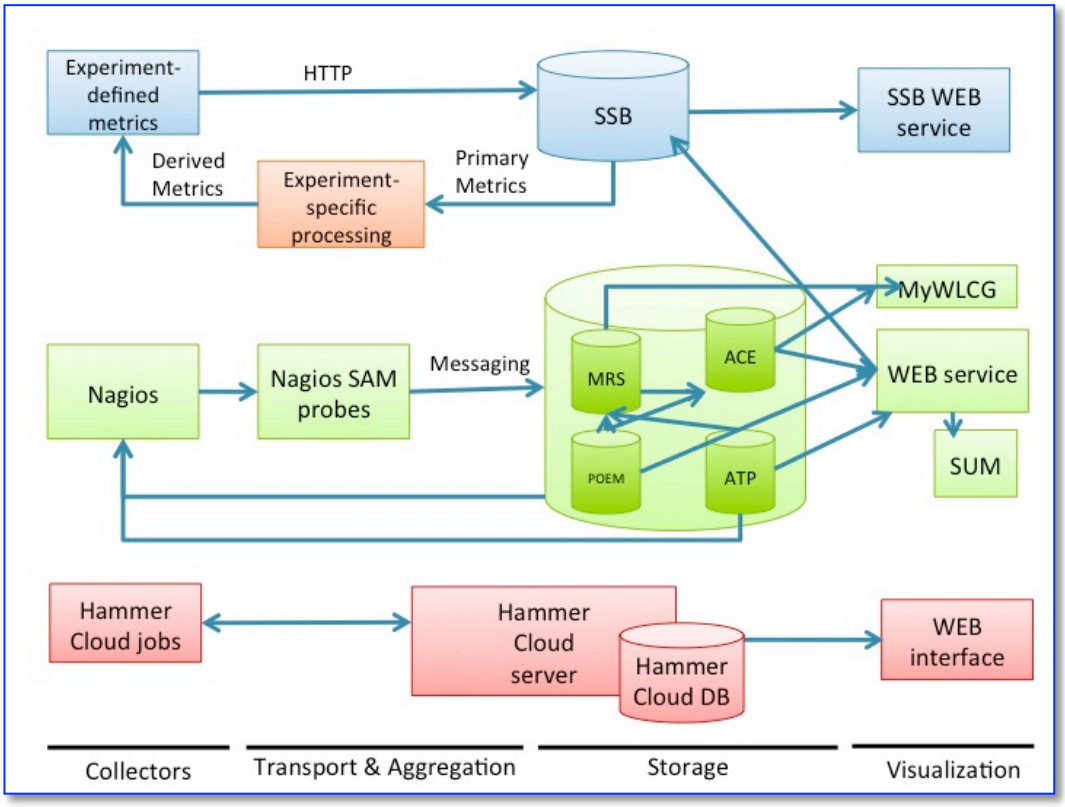

Figure 1. Architecture and design of the monitoring systems before the WLCG Monitoring Consolidation in the area of infrastructure monitoring.

The figure shows several limitations of the initial system:

1. There were three independent applications, which offer overlapping functionality. They also had to handle similar issues, like evaluating the topology of the infrastructure.
2. Some applications offered different API for exporting data and for its native web interfaces.
3. The information from one of the application might be relevant to the others; at the time, the only possible solution was to duplicate the data.
4. Several components talk directly to the databases of different components.

The deployment and support of each of the applications was done in a different way.

3. Proposed architecture

The architecture proposed for all the monitoring applications is depicted in Figure 2. It is worth pointing out that most of the monitoring applications provided by the group already followed that approach, and the effort was put into ensuring that all of them follow the same structure.

This layered design offers a great level of flexibility:

1. No dependencies between layers in terms of data formats and storage.
2. Components communicate only via APIs.
3. Different alternatives can be experimented for each layer without impact on the other layers.

For instance, it allows evaluating different visualization options, based on an existing storage layer. To allow this flexibility, it is very important that the layers communicate through well-defined APIs and data formats.

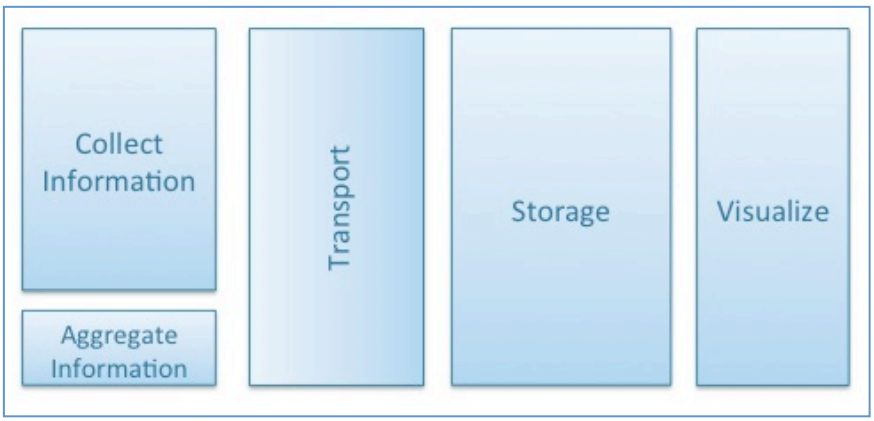

Figure 2. Components of the layered architecture

With this layered design, the structure of the Service and Infrastructure monitoring application can look like the Figure 3.

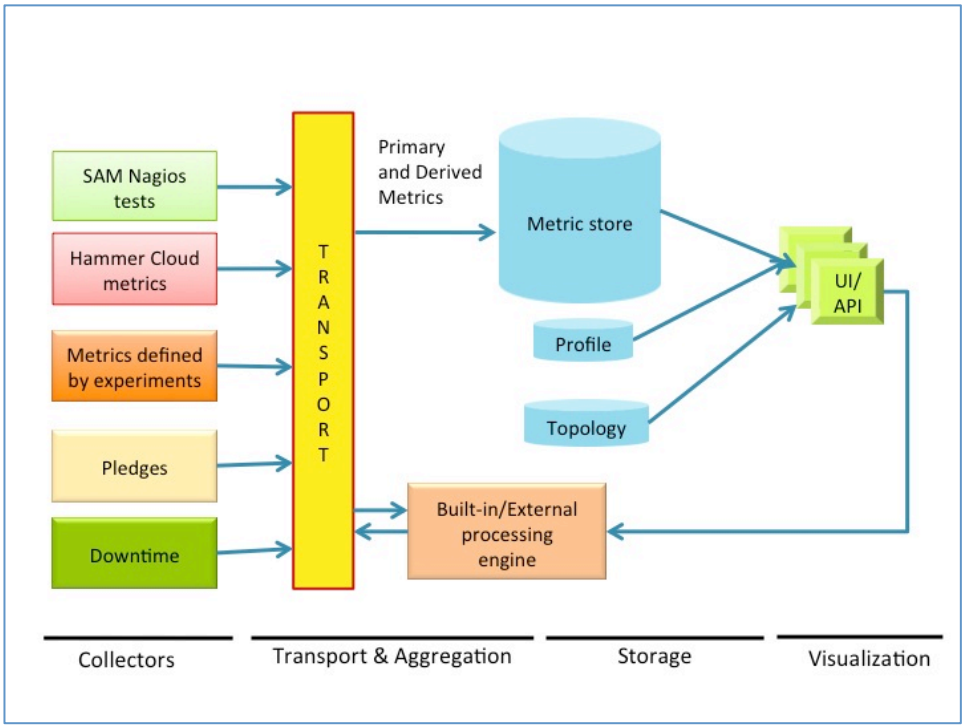

Figure 3. The layered design applied to the Monitoring Infrastructure

There are multiple producers of information on the left hand side. All of them publish the information to the transport in a predefined format. The information is then stored in a common metric store.

There are other databases containing the topology of the experiments and the profiles.

The aggregation is taken out of the storage and the transport. There are some generic mechanisms to combine metrics and to register them again in the metric store. Using those generic mechanisms, it is also possible to calculate the availability and reliability of the services and sites. On top of that, the experiments have the possibility of getting data out of the metric store (using the predefined API), apply their own algorithm, and feed the results back into the system.

The rest of this chapter describes each of the layers depicted in figure 2, first introducing the initial status and then the agreed solution.

3.1. Deployment and support

The deployment and management of services was done in different ways, on a case-by-case basis. Most of the machines were based on Quattor {REFERENCE}, with manual intervention in multiple machines, and different ways of managing the configuration: Quattor templates, sindes, yaim, etc.

The deployment took advantage of the new developments done in the Agile Infrastructure project.

It is beneficial to use a standard way for the management of the entire cluster, following the lead of the CERN IT department, which means using OpenStack, puppet, foreman and hiera.

All the machines needed for the monitoring infrastructure are under the control of a single group. This means that the support of the platforms can be reduced also to the CERN provided platforms. In practical terms, this means that all the machines used by the group for this purpose were moved to SLC6.

Another topic related to the deployment is the way applications are built and packaged. Before the working group was created, some of the applications used the Dashboard Build system (an in-house developed framework built on top of the python distutils and its own yum repository), and other applications used the EGI Koji. During the WLCG Monitoring consolidation project, all the applications supported by the team converged on a common procedure, aligned with the strategy of the rest of CERN IT: the CERN Koji.

The last topic on this layer is the version control mechanism. Initially, there were several SVN repositories that contain different applications. All of them were moved to git.

3.2. Collectors and probes

Before the creation of this group, the standard way of running probes was to configure them in Nagios. The four experiments had already defined a set of probes that test service functionality. The submission of all the worker nodes probes was done using WMS.

Experiments could inject their own metrics in the experiment “nagioses”, as long as the metrics and services were properly defined in multiple places.

Besides all the experiment tests, the services were tested with the credentials of the OPS virtual organization. Some sites had configured high priority queues to handle these tests. Initially, the official monthly reports with site availabilities were based on the results of OPS tests.

The consensus was that Nagios should stay as an optional way of submitting remote tests. There was some evolution needed on the Nagios probes, in order to submit through Condor-G and to complement the direct CREAMCE submission probes with the worker node test payloads.

At the same time, it was very important to incorporate metrics produced by other sources, like for example HammerCloud, DIRAC [9] or MonALISA[10]. The injection should be done at the transport layer, without having to pass by the Nagios box. The injected metrics could be defined either at the service level, or at the site level. This would allow also the submission of metrics from the standard workflows of the experiment, instead of depending on tests submitted through different channels. There should be an easy way of adding/removing metrics, sites and services.

OPS tests were not needed anymore for WLCG monitoring.

3.3. Transport

This layer decouples the information providers and the storage, and enables asynchronous communication between producers and consumers of the monitoring data. This approach facilitates data access from multiple consumers, and it endures better fault-tolerance and scalability. Initially, there were three different transport modes used (Messaging, HTTP put/get and MonALISA UDP).

The agreement was that no extra effort was needed in this area, since the existing transport mechanisms were well known. Each mode is best suited for a different use case, and the approach seemed to be adequate for the foreseeable future.

3.4. Storage

For historical reasons, there were too many different schemas, even if the data structure was very similar. Moreover, some of the applications interacted with each other through shared database tables, instead of using well-defined APIs. This created too many dependencies between the applications, and made it impossible to introduce changes in any component without touching the whole chain.

A more modular approach was recommended, where each component presents an API that will be used by other components. This facilitates the replacement of modification of components and the evaluation of alternative implementations.

Considering functionality, two of the Infrastructure Monitoring solutions mentioned beforehand (SSB and SAM) were very similar: given a set of instances and metrics, record their status evolution over time. At the same time, the schemas were very different, thus needing more effort to maintain and evolve both systems. The approach used in SSB was more generic, allowing the instances to be services, sites or even channels. It also simplified the definition of new metrics, and gave the possibility of combining metrics into views. Moreover, since the SSB stored by default only the status changes of the metrics (instead of every single value), the size of the database stayed under control. For all these reasons, it was decided to use the SSB schema to store also the SAM data. This way, the number of schemas maintained by the group decreases, and it even offers more functionality, since it allows the combination of SAM metrics with any other experiment-defined metrics.

3.5. DB Back-end technologies

All the applications based at CERN were currently running on top of ORACLE, and for some of them, MySQL was also supported.

During the project, there was an investigation for alternatives to ORACLE, in particular NoSQL solutions [11]. The group evaluated Hadoop/HBase and ElasticSearch. Two testbeds, one with Hadoop/HBase and the other with ElasticSearch, were instantiated and evaluated for different monitoring components. At the time of the project, the available version of ElasticSearch did not offer all the required functionality (in particular, grouping by multiple fields). A new version of ElasticSearch was going to address this issue. Therefore, it was decided to do the evaluation once again once the new version became available.

3.6. Aggregation

There were two main common use cases for data processing:

- The first one was processing the current value of a set of metrics, and as a result, producing a new metric. For example, the service status is calculated combining status of multiple service instances.
- The second case is the aggregation of metric results over a given time interval. For example, calculating the average transfer throughput based on set of distinct measurements.

In both cases a new metric is produced either as a result of processing of set of other metrics or aggregating a single metric or several metrics over time.

The output of the aggregation layer should be fed back into the system as a new metric, which could even be further included in the next processing iterations. This way, the same visualization layer as for the basic metrics can be reused, and, on top of that, the aggregated metrics can be as well aggregated even further.

3.7. Visualization and API

For historical reasons, there were plenty of differences across the applications:

- Different backend frameworks: Django, Dashboard
- Some applications request data through AJAX, some others request rendered html fragments
- Data tables are not used consistently across all tabular views

- Model-View-Controller (MVC) framework is not used consistently, or not used at all sometimes
- Plots come from different sources: Google Image Charts API, Google Charts API, Highcharts, Graphtool, Raphael
- Content is rendered using different technologies: XSLT, HTML skeleton, Django templates

It was recommended to have a separation between the server side and the client side as the main design principle for the visualization of the WLCG monitoring applications[12]. The server side should produce basic html skeleton pages and a REST API with json for AJAX requests.

The client part would be composed of JavaScript libraries, following an MVC paradigm. Open source client-side MVC frameworks were evaluated for existing and future applications.

A common look and feel was enforced where relevant. This is already the case for all the transfer monitoring applications (WLCG transfer Dashboard, ATLAS DDM Dashboard, xrootd dashboards [13]), accounting applications (ATLAS and CMS historical views, ATLAS DDM accounting portal [14]).

All the information available for the visualization layer was available in machine-readable format.

4. Achievements of the WLCG Monitoring Consolidation

The most important accomplishments of the project were:

- A substantial reduction of the effort for the Infrastructure Monitoring support. The effort dropped from 6 FTEs to less than 1 FTE.
- Convergence of the two infrastructure monitoring systems into a single one, thus reducing the effort needed to support them.
- Enforcement of a modular design for all the Infrastructure Monitoring applications. This facilitates investigating new technologies to replace some of the existing layers.
- A decoupled EGI [8] and WLCG monitoring systems, passing on the responsibility of some of the components to the EGI partners.
- Moving the generation of the monthly and quarterly reports from SAM to SAM3.
- The use of a common mechanism for the deployment, configuration and maintenance of the services. The machines used for the WLCG Monitoring applications were moved to virtual machines, running the latest operating system supported at CERN at the time.

5. Further evolution

The group identified several areas where more effort would be required. They can be divided in three categories:

- Taking advantage of the new functionality. The new infrastructure monitoring offers even more flexibility than the previous version. ATLAS and ALICE have already started to use this to get a better indication of the status of the services and sites. The effort in this area should continue to identify the critical metrics, and to combine them in the correct way.
- Improving the Nagios framework. The framework used for the submission of the tests needs to be reviewed and improved with the latest changes in the technologies.
- Evaluation of new technologies. It is very important to keep up to date with the latest developments in different areas. In particular, the storage, data processing and visualization depend heavily on external products. A very promising solution inspired by the so-called Lambda-architecture is being currently implemented for the WLCG Data transfer applications

6. Conclusion

In eighteen months, the WLCG Monitoring Consolidation achieved a great reduction of the effort required to maintain the WLCG monitoring applications. On top of that, it paved the way for a further evolution of the system, introducing a layered design and therefore allowing the replacement of components while minimizing the impact on the rest of the system.

The WLCG Monitoring Consolidation benefited from other projects that were going on at the same time. In particular, and thanks to the CERN Agile Infrastructure, all the machines and services needed for the WLCG Monitoring were deployed and configured in a consistent way.

The evaluation of new technologies is a continuous effort. The transition to the Lambda-architecture, which is being performed for the WLCG transfer monitoring applications, is also foreseen for other WLCG monitoring areas.

References

[1] Bird I, *"Computing for the Large Hadron Collider"*, 2011, *Annual Review of Nuclear and Particle Science*, 61:99–118 doi:10.1146/annurev-nucl-102010-130059

[2] Andreeva J et al., "Experiment Dashboard - a generic, scalable solution for monitoring of the LHC computing activities, distributed sites and services", 2012, J. Phys.: Conf. Ser. 396 032093 doi:10.1088/1742-6596/396/3/032093

[3] P Andrade et al. , *"Service Availability Monitoring Framework Based On Commodity Software"*, 2012, J. Phys.: Conf. Ser. 396 032008 doi:10.1088/1742-6596/396/3/032008

[4] P Andrade et al., *"Review of CERN Data Centre Infrastructure"*, 2012, J. Phys.: Conf. Ser. 396 042002 doi:10.1088/1742-6596/396/4/042002

[5] D C van der Ster *et al.*, *"HammerCloud: A Stress Testing System for Distributed Analysis"*, 2011, *J. Phys.: Conf. Ser.* 331 072036 doi:10.1088/1742-6596/331/7/072036

[6] I Dzhunov et al., *"Towards a centralized Grid Speedometer"*, 2014 *J. Phys.: Conf. Ser.* 513 032028 doi:10.1088/1742-6596/513/3/032028

[7] M Jouvin, *"Quattor: managing (complex) grid sites"*, 2008 *J. Phys.: Conf. Ser.* 119 052021 doi:10.1088/1742-6596/119/5/052021

[8] T Antoni *et al.*, "Sustainable support for WLCG through the EGI distributed infrastructure", 2011 *J. Phys.: Conf. Ser.* 331 082009 doi:10.1088/1742-6596/331/8/082009

[9] A Casajus et al,. *"DIRAC pilot framework and the DIRAC Workload Management System"*, 2010, J Phys.: Conf *Ser.* 219 062049 doi:10.1088/1742-6596/219/6/062049

[10] I Legrand *et al.*, *"Workflow management in large distributed systems"* 2011 *J. Phys.: Conf. Ser.* 331 072022 doi:10.1088/1742-6596/331/7/072022

[11] E Karavakis et al., *"Processing of the WLCG monitoring data using NoSQL"*, 2014, J. Phys.: Conf. Ser. 513 032048 doi:10.1088/1742-6596/513/3/032048

[12] D Tuckett et al., *"Designing and developing portable large-scale JavaScript web applications within the Experiment Dashboard framework"*, 2012, J. Phys.: Conf. Ser. 396 052069 doi:10.1088/1742-6596/396/5/052069

[13] J Andreeva et al., *"WLCG Transfers Dashboard: a Unified Monitoring Tool for Heterogeneous Data Transfers"*, 2014, *J. Phys.: Conf. Ser.* 513 032005 doi:10.1088/1742-6596/513/3/032005

[14] E Karavakis et al., *"Common Accounting System for Monitoring the ATLAS Distributed Computing Resources"*, 2014, J. Phys.: Conf. Ser. 513 062024 doi:10.1088/1742-6596/513/6/062024

[15] L Magnoni et al. *"Monitoring WLCG with lambda-architecure: a new scalable data store and analytics platform for monitoring at petabyte scale"*, proceedings of CHEP 2015, to be published by IOP Phy.: Conf. Ser.