Testing and integrating the WLCG/EGEE middleware in the LHC computing

S Campana[1], A Di Girolamo[1,2], E Lanciotti[1,2], P Méndez Lorenzo[1], N Magini[1,2], V Miccio[1,2], R Santinelli[1], A Sciabà[1]

[1] European Laboratory for Particle Physics (CERN), CH-1211 Geneva 23, Switzerland
[2] INFN – CNAF, Viale Berti Pichat 6/2, 40127 Bologna, Italy

E-mail: Simone.Campana@cern.ch, Alessandro.Di.Girolamo@cern.ch, Elisa.Lanciotti@cern.ch, Nicolo.Magini@cern.ch, Patricia.Mendez@cern.ch, Vincenzo.Miccio@cern.ch, Roberto.Santinelli@cern.ch, Andrea.Sciaba@cern.ch

Abstract. The main goal of the Experiment Integration and Support (EIS) team in WLCG is to help the LHC experiments with using proficiently the gLite middleware as part of their computing framework. This contribution gives an overview of the activities of the EIS team and focuses on a few of them particularly important for the experiments. One activity is the evaluation of the gLite workload management system (WMS) to assess its adequacy for the needs of the LHC computing in terms of functionality, reliability and scalability. We describe how the experiment requirements were mapped to validation criteria and how the WMS performances were accurately measured under realistic load conditions over prolonged periods of time. Another activity is the integration of the Service Availability Monitoring system (SAM) with the experiment monitoring framework. The SAM system is widely used in the EGEE operations to identify malfunctions in Grid services, but it can be adapted to perform the same function on experiment-specific services. We describe how this has been done for the LHC experiments, which are now using SAM as part of their operations.

1. Introduction

The Experiment Integration Support (EIS) team in the Worldwide LHC Computing Grid project (WLCG) was formed in 2002 to help the LHC experiments and other user communities to use the Grid as effectively as possible. The EIS activities are varied and include:

- contributing to integrate the experiment computing framework with the Grid middleware;
- interfacing user communities with the middleware developers and the WLCG infrastructure operations;
- developing new user tools to implement functionalities missing from the Grid middleware;
- testing middleware components as they become available;
- directly participating to the experiment computing activities (computing challenges, Monte Carlo production, etc.);
- providing end-user documentation.

This contribution describes some of the recent activities of the EIS team which had a particularly significant contribution towards improving the level of integration between the

experiment computing systems and the WLCG grid. The first activity consisted in thoroughly testing the gLite workload management system (WMS) [1], which allows to submit and manage Grid jobs by choosing the best resource matching the job requirements. The second activity is the integration of part of the experiment monitoring with the SAM framework [2].

2. Testing the gLite Workload Management System

The LHC experiments need to generate large amounts of simulated data to validate their reconstruction software, test the computing model and develop physics data analysis algorithms. In 2007, the largest collaborations, ATLAS and CMS, have produced up to the order of 50 millions of simulated events per month and this volume will double in 2008. Data analysis is even more demanding, at least in terms of numbers of submitted jobs, as each experiment is expected to be submitting a few hundreds of thousands of jobs every day in 2008.

The gLite Workload Management System (WMS) [1] is a relatively new component of the gLite middleware stack and it will replace the LCG Resource Broker as the system to dispatch and manage jobs submitted to the WLCG/EGEE Grid infrastructure. The LCG RB is sufficiently reliable for production purposes, but it is not able to reach submission rates exceeding a few thousands of jobs per day. The WMS, on the other hand, has some features that make it much more scalable, in particular:

- support for "bulk" submission of jobs via job collections;
- support for "bulk matchmaking" of collections;
- efficient transfer of sandbox files shared among different jobs in a collection.

Other features increase the stability and the reliability of the system, in particular a mechanism to refuse new jobs in case the system load exceeds a given threshold and the automatic resubmission of failed jobs.

To assess the usability of the WMS as a component of the experiment computing system, the requisites of ATLAS and CMS were collected, and summarized by WLCG as a single set of criteria to decide on the readiness of the WMS. These criteria are shown in table 1.

Table 1. Acceptance criteria for the gLite WMS.

Year	CMS	ATLAS	WLCG
		Performance	
2007 2008	50,000 jobs/day 120,000 jobs/day using ≤ 10 WMS instances	40,000 jobs/day 100,000 jobs/day using ≤ 10 WMS instances	10,000 jobs/day
		Stability	
		$\leq$ 1 restarts of WMS or LB per month under load	no performance degradation or need to restart services for $\geq$ 5 days; $\leq 1\%$ of stale jobs

To ensure that the WMS could meet the requisites in time, starting from July 2006 the testing of the system was performed by the EIS team bypassing the testing and certification

procedure and in close interaction with the developers. This process involved a fast loop of testing, bug discovery and patching. The test was twofold: one part of it consisted in submitting large numbers of "hello world" jobs with requirements similar to those of real experiment jobs; another part consisted in using the WMS for real Monte Carlo production in ATLAS [3]. CMS also used the gLite WMS to submit part of the analysis jobs during the CSA06 challenge [4].

One of the most relevant problems found during the testing was the poor stability when submitting job large collections due to an implementation based on DAGMan, a meta-scheduler for Condor [5] used to describe dependencies between jobs; the stability and performance of the WMS substantially increased after reimplementing collections as simple sets of uncorrelated jobs. Also the memory consumption was greatly reduced, by reducing both the number of parallel threads and the memory used by some of the WMS services. After several bug fixes and improvements, the job failure rate due to the WMS for submission rates of about 15,000 jobs/day decreased from 15% to less than 1%. Finally, as a way to prevent the degradation of the WMS performance over time when heavily used, a limiter mechanism was implemented to prevent the submission of new jobs if the load exceeds a certain threshold.

In April 2007, a test to finally demonstrate if the gLite WMS could satisfy the WLCG acceptance criteria was run. These were the results:

- 115,000 jobs were submitted during 7 days (16,000 jobs/day), using collections of 100 jobs each;
- the Condor queue of the WMS contained about 10,000 jobs for the duration of the test;
- 320 jobs were not normally processed, due to a timeout in the communication with the LB server;
- the limiter prevented the submission of 1,500 jobs/day because the load on the WMS machine exceeded a threshold of 10.

In figure 1 the number of jobs in each possible status is shown: the number of jobs in "transient" states remains negligible, while the number of scheduled and running jobs reaches a constant value, as expected. These results allowed to consider the acceptance test passed. Figure 2 shows that the time delay between the job submission, the job assignment to a CE by the WMS and the submission to the CE batch system is always negligible.

Another set of tests was conducted to measure the performance of the gLite WMS when submitting single jobs instead of collections. Increasing the number of parallel submitting processes up to 12, the submission rate was seen to saturate at about 20,000 jobs/day.

3. Integration of the experiment monitoring system with the Service Availability Monitoring

3.1. The SAM framework

The Service Availability Monitoring System (SAM) [2] is a framework developed in EGEE to provide a global and uniform monitoring tool for Grid services. It works by executing periodic tests, provided by "sensors" (one for each type of Grid service), on all the Grid service instances known to the Grid Information System. The test results are published in an Oracle database with a Tomcat-based web service interface. The information can be retrieved from the SAM database by means of HTTP queries via a programmatic interface. SAM is one of the main sources of information for Grid operations and is used to measure the availability of Grid services.

The flexibility of the SAM framework is essential to allow also for any Virtual Organization to implement custom tests on existing service types, or even on VO-specific services. A clear advantage is the possibility to use the same SAM database and visualization tools already developed for the EGEE operations.

The following sections describe how the LHC experiments are using SAM to perform part of their monitoring.

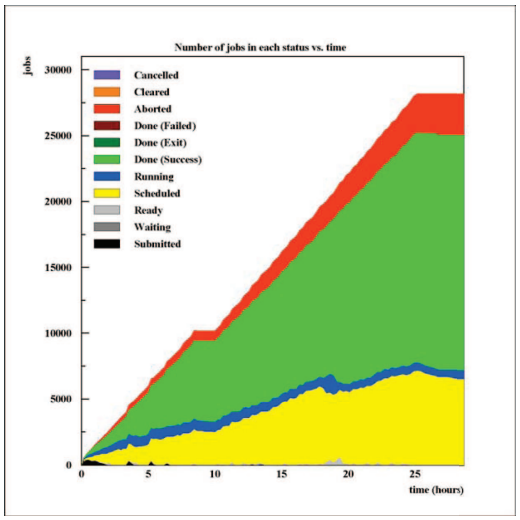

Figure 1. Number of jobs in each status as a function of time during submission.

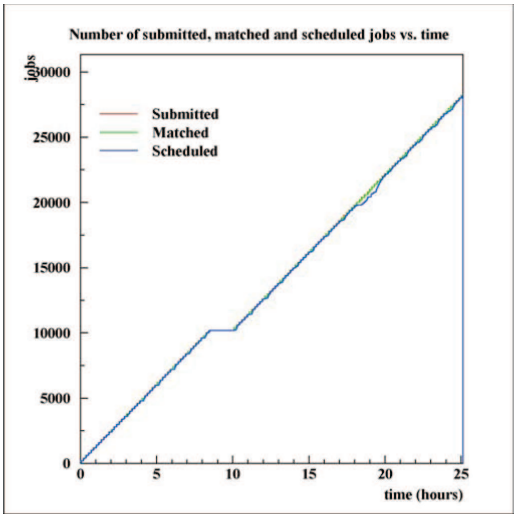

Figure 2. Number of jobs submitted, matched to a CE and scheduled as a function of time.

3.2. ALICE

3.2.1. VOBOX monitoring The VOBOX is a Grid service provided by WLCG to the LHC experiments with the purpose of hosting experiment-specific services and agents and to provide automatic proxy renewal functionalities and a GSI-SSH server. The shared file system typically used to install the experiment software on a Grid site is also accessible from the VOBOX. Access to the VOBOX is restricted to the Software Group Managers (SGM) of the experiment.

The ALICE computing system [6] requires a VOBOX to be deployed at each site, to submit and manage long-lived pilot jobs to the Grid and to install the ALICE software. There are approximately sixty VOBOX instances dedicated to ALICE in WLCG. The monitoring of this service is required to ensure a smooth and optimal usage of the available resources. It has been decided to use the SAM framework as the tool to implement the VOBOX monitoring. There are three basic requisites for the VOBOX monitoring with SAM:

(i) the possibility to define and change at any time the tests to be run on the VOBOX;
(ii) the possibility to maintain a list of the VOBOX instances to be tested;
(iii) failures of critical tests should automatically trigger alarms.

3.2.2. Implementation of the ALICE SAM tests The test suite developed for the ALICE VOBOX consists of several tests, which perform checks on the following items:

- the functionality of the proxy renewal service, the procedure to register user proxies and the duration of the renewed proxies;
- the permissions of the software installation area;
- the status of the locally submitted jobs;
- the status of the Resource Broker used to submit jobs.

The tests are run from a SAM user interface located at CERN and managed by ALICE; a script which executes the tests on every VOBOX is run as a cron job every two hours and collects the test results, which are published to the SAM database. The script will also read

the list of VOBOX instances maintained by ALICE and update accordingly the SAM database in case VOBOX instances have been added or removed. The possibility to insert in the SAM database service information taken from an external file accessible via HTTP was specifically developed for ALICE.

Finally, the SAM team has provided ALICE with an alarm system able to send notifications via email or SMS in case of failures of critical tests. The notifications are sent to the person responsible for the VOBOX instance.

3.3. CMS

The approach chosen by CMS to the integration of SAM in the CMS monitoring system consists in writing specific tests and run them using the standard SAM sensors, in particular the one for the CE and the one for the SRM service. The test results are then used to measure the "quality" of these services at the different CMS sites.

The general goal of these tests is to discover all possible problems that could make a CMS job to fail, in particular those related to the local installation and configuration of the CMS software and to any CMS-specific site services.

3.3.1. The Computing Element tests The CMS tests for the CE perform several checks on the local installation of the CMS software, on some site services (the local FroNTier cache of the calibration and alignment database [7]) and on some operations, like a file transfer from a worker node to the local storage system. The tests are summarized in table 2.

Table 2. CMS tests for the computing element.

Test name	Test definition
js	Checks that it is possible to submit a job to the CE
basic	Checks the CMS software area and the local site configuration
swinst	Checks the installed versions of CMSSW
mc	Tries to copy of a file from the worker node to the local SE
frontier	Reads calibration data using CMSSW via the local Squid server
squid	Makes a simple query to the local Squid server

The submission of the CE tests is done using two different VOMS roles [8]: one, called *lcgadmin*, which allows to benefit from a higher job priority at most WLCG sites and one, called *production*, which is the same used to run Monte Carlo production jobs; the former is used to run all the CE tests, apart from the *mc* test (table 2), which is run with the latter, to ensure that the storage system is accessed with the same privileges of a Monte Carlo production job.

All the jobs are submitted via an LCG Resource Broker to all CMS sites, inclusing those which are part of the OSG project, which uses a different middleware compared to EGEE. The ability of the RB to submit to both grids was expected, due to recent efforts to improve the level of interoperability between them.

3.3.2. The SRM tests The CMS tests for the Storage Resource Management (SRM) services [9] check if the SRM is available and accessible for basic data transfer and management operations: copying a file to the SRM (*put*), verifying file metadata on the SRM, copying a file from the SRM (*get*), deleting the file. The tests, summarized in table 3, are run sequentially with the

production VOMS role to ensure that the same credentials used for Monte Carlo production are used.

Table 3. CMS tests for the Storage Resource Management service.

Test name	Test definition
get-pfn-from-tfc	perform lfn to pfn matching
put	Store file to SRM (put)
get-metadata	Verify file stored in SRM
get	Copy a file back from the SRM (get)
advisory-delete	Delete a file from the SRM

As a preliminary step, the file to transfer is created and its size and checksum are written. The first test chooses a logical file name (LFN) for the file to transfer and determines the corresponding physical file name (PFN) on the target storage element. CMS does not employ database-based global catalogues to determine the association of logical to physical file names; at each site a trivial file catalogue (TFC), a text file with a handful of mapping rules maintained by the site operators, is used to determine actual file locations. The test retrieves the TFC for the site and associates a PFN to the chosen LFN according to the rules therein.

Then, the *put* test verifies the ability to copy the file to the SRM. The SRM client, *srmcp*, included in the gLite distribution, is used to prepare and execute the transfer using the GridFTP protocol.

The next test uses the SRM client to retrieve the metadata of the file stored remotely on the SRM and verify that size and (if returned by the SRM) checksum match with those of the original local copy.

The *get* test checks if it is possible to retrieve a copy of the remote file from the SRM. The SRM client is used to prepare and carry on the transfer using the GridFTP protocol and the file transferred back is then compared with the original.

The final test issues a command of advisory deletion on the remotely stored file. Since advisory deletion is not required to be synchronous, the test only verifies that the command was successful, without checking if the remote file was actually deleted or not.

The SRM tests developed for CMS were also used for the ATLAS SAM tests with only minor modifications.

3.3.3. Site readiness The CMS SAM tests for the computing element are currently used as a way to measure the likelihood to encounter problems when running CMS jobs at a site. For each site, the fraction of successful tests over the total number of performed tests during the day is calculated and it is plotted as a function of time (figure 3). Sites are urged to periodically check the results of the SAM tests and to act on any failure. An overall view of the CMS site readiness is obtained by averaging the site quality estimation over a longer period of time (figure 4).

3.4. ATLAS

The ATLAS experiment is developing SAM tests to monitor the availability of computing elements, storage elements and SRM services, and to verify the proper functioning of the ATLAS software installation, in a way very similar to CMS.

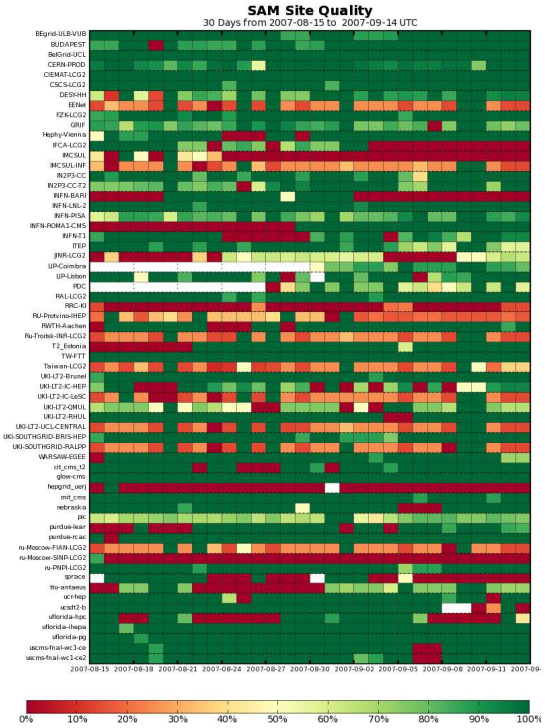

Figure 3. Daily site quality estimation for the CMS sites from 15-8-2007 to 14-9-2007.

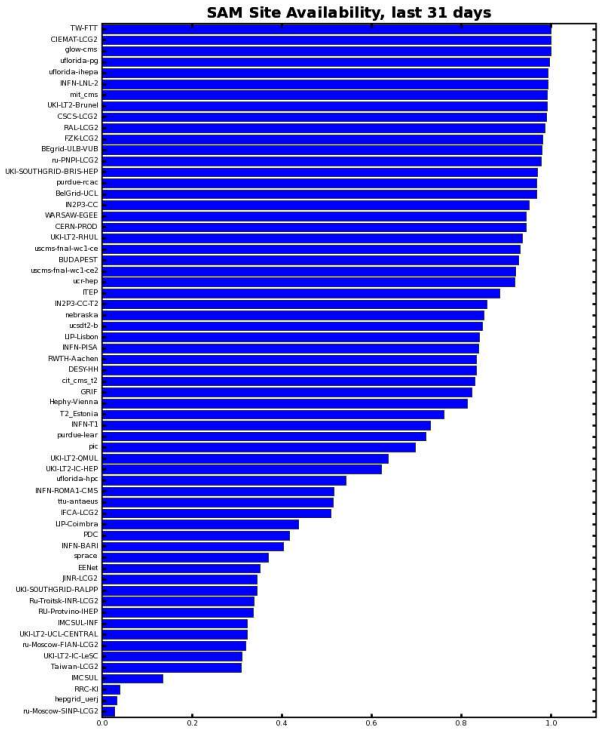

Figure 4. Average site quality estimation for the CMS sites from 15-8-2007 to 14-9-2007.

Concerning the SRM tests, to take into account the fact that there may be different SRM endpoints to test for a single node, a file containing the definition of the endpoints relevant to ATLAS is used. Different endpoints belonging to the same node might need to be tested using different VOMS groups or roles, if access permissions are restricted to specific VOMS groups or roles.

For each SRM endpoint, tests are run to *a)* write a small file to the remote storage system, *b)* verify the metadata information, *c)* copy the file back and compare it to the original and *d)* delete the remote file.

In order to validate the software installation of the ATLAS software on the computing element, SAM is used to send jobs that use the ATLAS software to analyze a very small number of simulated events.

Finally, ATLAS plans to make use of the notification system developed in SAM to alert the relevant people via email or SMS about failures of critical tests.

3.5. LHCb

SAM is currently used by LHCb to constantly monitor the health status of site batch farms and consequently to compute the availability of critical site services (CE and SRM endpoints) as measured by experiment-specific tests. The same job slots are also used for the installation of the LHCb software and for its validation by running tests that emulate the full LHCb Monte Carlo production chain.

In order to allow SAM jobs to be executed as soon as possible, they are submitted with a VOMS role which is granted a high priority on the batch system. In addition to that, a certain number of special SRM tests are executed only on Tier-1 sites; in particular, some of these

checks are gathering information from an LHCb service called *stager agent*, which is in charge of pre-staging files on remote SRM endpoints, to allow LCHb reprocessing tasks to find on disk the required input files. As a consequence, the whole daily production activity contributes to rate a site.

Early experiences of LHCb with SAM highlighted some limitations of the SAM framework as far as concerns the management of grid jobs for CE tests. These limitations, also recognized by SAM developers, are maily due to the rigidity of the SAM infrastructure and make difficult to debug problems [10].

A way to circumvent these limitations was to use DIRAC [11] to take care of the job management (job submission, job monitoring and job output retrieval); the publication of the test results, the computation of the site availability, the storage of historical information are instead done using the usual SAM tools and the central SAM database.

This hybrid solution is in production since July 2007 and it has been proved to be extremely reliable, as it combines the functionalities for detecting problems offered by the SAM framework with the benefits in using a reliable workload management system like DIRAC, which provides a more effective monitoring of the progress of Grid jobs.

4. Summary and conclusions

The middleware testing and integration continues to be a central part of the activity of the EIS team. In particular, it significantly contributed to bring the gLite WMS to a quality level which makes it usable in a production context. Both ATLAS and CMS are using the WMS to submit both Monte Carlo production and data analysis jobs at submission rates that reach 20,000 jobs/day. This testing process is still ongoing, and the EIS team actively participates in the experiment operations involving the use of the gLite WMS.

The integration of the experiment monitoring with the SAM framework used for the EGEE operations is also having a positive impact on the ability of the LHC experiments to identify and cure site problems not specifically related to the Grid middleware; for each experiment different solutions were developed to best adapt to their computing systems. CMS and ATLAS have chosen to integrate their specific tests to existing sensors, using the SAM framework also for the test submission; ALICE uses SAM to publish information about the status of all their VOBOX instances, while LHCb integrated the test submission with the DIRAC workload management, using also the information from normal jobs to assess the status of their computing resources. When necessary, missing functionalities were added to the SAM framework.

References

[1] Avellino G *et al* 2007 The gLite workload management system *these proceedings*

[2] Duarte A, Nyczyk P, Retico A and Vicinanza D 2007 Monitoring the EGEE/WLCG Grid services *these proceedings*

[3] Campana S, Rebatto D and Sciabà A 2007 Experience with the gLite workload management system in ATLAS Monte Carlo production on LCG *these proceedings*

[4] Gutsche O and Hajdu C 2007 WLCG scale testing during CMS data challenges *these proceedings*

[5] Silberstein M, Geiger D, Schuster A and Livny M 2006 *Proc. of the 15th IEEE Symp. on High Performance Distributed Computing (Paris)*

[6] Carminati F *et al* 2005 ALICE computing: technical design report *CERN-LHCC-2005-018*

[7] Blumenfeld B, Dykstra D, Lueking L and Wicklund E 2007 CMS conditions data access using FroNTier *these proceedings*

[8] Alfieri R *et al* 2004 *First European Across Grids Conference (Santiago de Compostela)* (Berlin: Springer) p 33

[9] Donno F *et al* 2007 Storage Resource Manager version 2.2: design, implementation and testing experience *these proceedings*

[10] Closier J 2007 Ensuring Grid resource availability with the SAM framework in LHCb *these proceedings*

[11] Tsaregotodtsev A *et al* DIRAC: a community Grid solution *these proceedings*