Journal of Physics: Conference Series **331** (2011) 072001 doi:10.1088/1742-6596/331/7/072001

Experiment Dashboard for Monitoring of the LHC Distributed Computing Systems

J. Andreeva, M. Devesas Campos, J. Tarragon Cros, B. Gaidioz, E. Karavakis, L. Kokoszkiewicz, E. Lanciotti, G. Maier, W. Ollivier, M. Nowotka, R. Rocha, T. Sadykov, P. Saiz, L. Sargsyan, I. Sidorova, D. Tuckett

CERN, European Organization for Nuclear Research, Switzerland

E-mail: Julia.Andreeva@cern.ch, Edward.Karavakis@cern.ch, Pablo.Saiz@cern.ch, David.Tuckett@cern.ch

Abstract. LHC experiments are currently taking collisions data. A distributed computing model chosen by the four main LHC experiments allows physicists to benefit from resources spread all over the world. The distributed model and the scale of LHC computing activities increase the level of complexity of middleware, and also the chances of possible failures or inefficiencies in involved components. In order to ensure the required performance and functionality of the LHC computing system, monitoring the status of the distributed sites and services as well as monitoring LHC computing activities are among the key factors. Over the last years, the Experiment Dashboard team has been working on a number of applications that facilitate the monitoring of different activities: including following up jobs, transfers, and also site and service availabilities. This presentation describes Experiment Dashboard applications used by the LHC experiments and experience gained during the first months of data taking.

1. Introduction

The Large Hadron Collider (LHC) [1] experiments chose for their computing systems a distributed tiered computing model. The LHC experiments rely on the Worldwide LHC Computing Grid (WLCG) [2] infrastructure which consists of more than 170 computing sites located all over the world. The LHC computing activities include data transfer and job processing. These activities are widely distributed and large scale. The LHC users submit more than 1M jobs per day and more that 100K jobs are running concurrently. More than 100 TB of data is transferred per day on the WLCG by the LHC experiments. Reliable monitoring of the state of the WLCG infrastructure and of the LHC computing activities is a vital condition of the overall success of LHC computing.

The Experiment Dashboard system aims to provide powerful and flexible monitoring of data transfer, job processing and the status of the distributed sites and services which perform the LHC computing tasks. The system enables following job and data transfer execution, detecting and investigating inefficiencies and failures, assisting in commissioning of sites and services, identifying trends, and predicting future requirements. The Experiment Dashboard works transparently across several middleware platforms. It is used by various categories of LHC users, among them physicists running their analysis on the Grid, people taking part in LHC computing shifts, user support teams, managers of various computing projects and site administrators. Many of the Experiment Dashboard applications are generic and shared by several LHC experiments, among them job monitoring, site usability and site status board applications. During the first year of data taking the Experiment Dashboard system proved to be an essential part of LHC computing and was widely used by the LHC experiments, particularly by ATLAS and CMS. For example, the CMS Dashboard server is accessed by up to 5K unique visitors (IP addresses) per month and more than 100K pages are viewed daily.

The next section of the paper describes the architecture of the Experiment Dashboard system. Sections 3 to 5 overview data transfer, job monitoring and site and service monitoring applications correspondingly and summarize the experience of using these applications during the first year of data taking. The final section contains our conclusions.

2. Architecture

Experiment Dashboard applications adhere to a set of core development principles: common technology and implementation, loose-coupling to data sources, sharing of monitoring data, and user involvement in the development process. The aims of these principles are to reduce development and maintenance overhead, allow applications to be easily adapted for use by multiple experiments, enable reuse of monitoring data, and ensure that applications meet user requirements. Where possible these principles are embodied in a common architecture and framework.

2.1. Common architecture

The architecture of Experiment Dashboard applications consists of some or all of the following components: data sources, consumers and collector agents, data repository, statistics agents, web user interfaces, machine-readable data interfaces. A schema of the architecture is presented in figure 1.

The data sources can be any external information system, such as external monitoring systems, grid services, experiment-specific central and distributed services, and specially instrumented jobs. Event data is stored in the data repository by consumers, which receive data via HTTP, MonALISA [3] or Messaging System for the Grid (MSG) [4], and by collector agents, which fetch data via SQL or HTTP. A data repository, normally an Oracle database, is used to store reference, event and statistics data. Statistics agents run periodically to compute statistics. Finally, the monitoring data is exposed via one or more web user interfaces, as well as in machine-readable format such as XML, CSV or JSON.

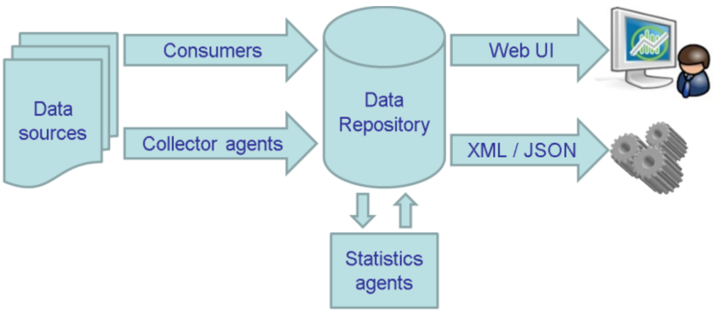

Figure 1. Experiment Dashboard common architecture.

2.2. Common framework

The Experiment Dashboard framework is implemented in Python and provides core components, libraries and tools, described below, which are reused by all Experiment Dashboard applications.

An automated build system extends Python distutils to enforce strict development, test and release procedures for the various common and application-specific components. An agent component includes libraries and tools to develop, manage and monitor agents. A data access component includes connection pooling, transaction handling and multiple data source configuration, which application-specific Data-Access-Objects (DAOs) use to abstract access to the data repository. A web framework handles HTTP requests. Following the Model-View-Controller (MVC) pattern, the controller maps URLs to application-specific action classes. The action classes retrieve data which the controller passes onto view classes. Generic view classes generate machine-readable responses as well as graph images. An HTML view class generates HTML from XML output based on application-specific XSL style sheets. The format and caching of the response can be easily configured for each action. CLI and API components provide base classes to simplify the development of Python API and command line access to the machine-readable responses provided by the server.

The framework continues to evolve to incorporate new technologies and reduce development and maintenance overhead. For example, recent developments take advantage of MSG for fast reliable data consumption and AJAX technologies for highly interactive web interfaces. Furthermore, improvements to the build system have streamlined application deployment and administration.

3. Dashboard for Data Transfer Monitoring

Experiment Dashboard includes two applications for data transfer monitoring: ALICE Transfer Efficiency Dashboard and ATLAS Distributed Data Management (DDM) Dashboard. This section provides details of ATLAS DDM Dashboard.

3.1. Functionality

The ATLAS computing model [5] requires the proximity of data to computing elements where jobs analyzing that data are executed. To support optimal utilization of the highly distributed resources of the grid, large amounts of data must be exported reliably and rapidly. The ATLAS DDM system, which is responsible for data distribution, consists of many components including catalogs for bookkeeping and distributed site services for coordinating data transfer requests. It is, by necessity, a complex system with high reliability and performance requirements. DDM Dashboard plays a key role in enabling the DDM system to meet those requirements.

The main web user interface provides access to statistics in plot and tabular format for user-defined periods. Statistics include dataset registrations, file transfers, and error message samples. Users can drill down to view individual dataset and file events including individual error messages. This interface typically has 1K unique visitors a month and 15K page views a day [6]. DDM Dashboard is used 24/7 by people taking computing shifts to identify and investigate failures and alert sites. This is an example of how DDM Dashboard is used to maintain the high reliability of the DDM system.

In addition to the web user interface, statistics data is accessible in machine readable format, allowing it to be easily incorporated into external applications. In particular, the mean and variance of transfer rates between storage element endpoints is regularly accessed by the ATLAS DDM site service components to optimize transfer channel selection. This is an example of how DDM Dashboard directly contributes to maintaining the high performance of the DDM system.

3.2. Implementation

DDM Dashboard follows the common Dashboard architecture described in the previous section. A single data repository stores reference data, events and statistics. The main consumers of dataset and file transfer events are implemented as HTTP endpoints using the Dashboard web framework. Sufficient reliability is ensured by persistency on the client side of unsent messages. Site availability data is pulled hourly via HTTP from Site Availability Monitoring (SAM) [7] by a collector agent. Statistics agents compute statistic summaries into 10 minute and 24 hour bins.

3.3. Performance

The Scale Testing for the Experiment Programme '09 (STEP'09) [8] demonstrated the readiness of the WLCG infrastructure for the LHC data taking. DDM Dashboard successfully participated in STEP'09, where ATLAS achieved transfer rates of 6 GB s^{-1}. The performance of DDM Dashboard is limited by the number of transfer state events it can consume and process, which is proportional to the number rather than size of file transfers. Exceptionally, the ATLAS April 2010 reprocessing campaign peaked at over 1 million file transfers per day, more than 3 times STEP'09. As a consequence, DDM Dashboard statistics computation procedures were tuned to handle this increased load using parallelization and bulk SQL. Subsequently, DDM Dashboard has successfully handled a peak of 1.5 million file transfers per day and regularly handles half a million transfers per day [9].

3.4. Future developments
A new web front-end is currently under development based on AJAX technologies, with the aim of making it easier to identify the location of failures. The key feature is an interactive matrix of transfer statistics showing both source and destination, with links to error samples and transfer details.

4. Dashboard for Job Monitoring
The Experiment Dashboard for job monitoring provides a consistent and complete picture of the job processing activity regardless of the Grid middleware or of the Virtual Organization (VO)-specific workload management system and it can also be used by VOs outside the LHC and High Energy Physics (HEP) community [10].

The jobs are instrumented to report their execution status to the Experiment Dashboard in real-time and the job submission tools of the VOs and the job wrappers generated by these tools are also instrumented to report task[1] meta-information, such as the creation time of the task, the input data collections and the number of events to be processed by each job. Information can be transported asynchronously [11] from the data sources with MonALISA or MSG. In both cases common libraries are provided to enable the instrumentation of VO-specific frameworks for Dashboard reporting. In addition to the instrumentation of the jobs, the Experiment Dashboard also imports data from the PanDA system [12] for the ATLAS experiment. The data is collected in the central repository and a set of distinct user interfaces aimed at different categories of users is built on top of that repository.

The Experiment Dashboard generates weekly reports with monitoring metrics related to the data analysis on the Grid; these metrics include the number of users, processed jobs, used slots, success rate and the most common reasons of failure. Based on these reports, the CMS analysis support team takes actions in order to improve the success rate of the user analysis. Every substantial negative fluctuation of the success rate is investigated and addressed [13].

The job monitoring applications described below are used extensively within the CMS community, with more than 200 distinct daily users, and have recently been ported to the ATLAS VO.

4.1. Generic job monitoring
The goal of the generic job monitoring is to follow the job processing of the LHC experiments on the distributed infrastructure by providing a very flexible access to recent monitoring data. The entry point of the application is the number of jobs submitted or terminated in a chosen time period categorized by their activity such as analysis, production or testing. The application provides a wide flexibility to users by allowing them to drill down the available information, expanding the set of jobs by various relevant properties such as execution site, grid gateway, user and completion status. This application is used by various categories of users, including site administrators. It provides all necessary details about job processing for a particular VO at a site, allowing, for example, detection of problematic execution nodes and identification of which input data set is accessed by a particular job.

4.2. Monitoring of user analysis activities
The Task Monitoring application [14] exposes a user-centric view of information to the user regarding submitted tasks. The information on a task includes the job status of individual jobs in the task, their distribution by site and over time, the reason of failure, the number of processed events and the resubmission history. The application offers a wide variety of graphical plots that visually assist the user to manage tasks by helping him identify any problematic site and blacklist it from further resubmissions. In the case of failure, the distribution by reason is provided, whether it be Grid-Aborted or Application-Failed jobs. Various kinds of consumed time plots are available that will help the user to see how the CPU time per event and efficiency can vary depending on the site that the jobs are running on. The monitoring tool has become highly critical among CMS users; up to 350 distinct analysis users are using it daily for their work.

[1] Here task is defined as a collection of jobs with the same input requirements.

Journal of Physics: Conference Series **331** (2011) 072001 doi:10.1088/1742-6596/331/7/072001

4.3. Accessing historical data for long-term statistics

The Historical View application offers job statistics distributed over time and it follows the evolution of the numeric metrics such as the number of jobs running in parallel, CPU and wall clock consumption or success and failure rates. The application offers a wide variety of plots that are useful for understanding how the job efficiency behaves over time, how resources are shared between different activities, and how various job failures fluctuate as a function of time. The distributions provided by this application are used by the LHC experiments in order to follow the usage of the infrastructure and the efficiency of data processing over time. For example, the plots are reviewed at the weekly CMS facilities and operations meetings in order to compare pledged resources against used resources. This application is also used by the CMS Data operations team to have a global view of the progress of CMS production jobs on the WLCG.

Another application for accessing historical data statistics is the Input Dataset Monitoring for CMS that offers a dataset-centric view such as the most accessed datasets over a selected period of time. The application is widely used by site administrators and the analysis support team and facilitates decision making about data replication or data deletion.

5. Dashboard for Site and Service Monitoring

Experiment Dashboard includes four different applications for site monitoring: Site Status Board, Dashboard SAM Web Portal, Siteview and the WLCG Google Earth Dashboard. This section will focus on the first two; the last two are described in a different article [15].

5.1. Site Status Board

The Site Status Board (SSB) is used to monitor the behavior of all the centers participating in the computing activities of a particular VO. It provides a single entry point that summarizes the status of the sites. The main idea is to provide a flexible framework which would allow VOs to define multiple monitoring metrics describing properties of sites and services and their performance, this includes definition of the content, criticality and update frequency of each particular metric and the way various metrics are combined and displayed. The metrics can be added, deleted and modified easily allowing VOs to quickly identify the most representative metrics. The most critical metrics can be combined into a single value for each site, corresponding to one of five statuses: working, warning, error, maintenance or under investigation. The SSB also keeps the history of how all the metrics have evolved over time. Moreover, it can also keep links to URLs with detailed information about the status of any metric.

The SSB was originally designed for the CMS offline shifters. Thanks to its flexibility, it was also adopted by the CMS Site Readiness program [16]. More recently, it has also been deployed for other experiments like ATLAS and LHCb.

5.2. Dashboard SAM Web Portal

SAM is an infrastructure that allows the execution of tests to validate the status of different grid services. Each VO can define its own tests.

Experiment Dashboard offers an application, the Dashboard SAM Web Portal, to monitor the results of these tests, and calculate and display the usability of sites based on these results. The application was developed from a VO perspective, therefore presenting the information in the most intuitive way for the VO administrator. It contains information about the last tests that were executed for each site and for each service, plus a historical view where the user can select the time range, sites and services that should be displayed. It also provides the possibility to combine the results of tests to calculate the availability and reliability of the different services and sites.

As with all Dashboard applications, the information can be retrieved in machine-readable format. It is being used by other monitoring projects like the CMS widgets [17].

The application is widely used by the four main LHC experiments. The site usability plots for Tier 1 sites are discussed at the daily WLCG operations meetings and weekly plots are included in the biweekly reports to the WLCG Management Board.

6. Conclusion

The Experiment Dashboard system has proved to become an important component of the LHC computing operations. Usage of the system by the LHC community in terms of the number of users, and the volume and frequency of monitoring data accessed, as well as the areas covered by the Dashboard applications are steadily growing. The system has been demonstrated to cope well with the load which is steadily increasing, in particular after the start of LHC data taking. The Experiment Dashboard has a positive impact on the WLCG infrastructure and user experience by promptly identifying job failures, which is particularly important for analysis users, providing feedback for tool optimization, as for example with the ATLAS Distributed Data Management system, assisting in site commissioning activities, and effective organization of LHC computing shifts. Involvement of the LHC user community in the development process and early validation of new applications by pilot users successfully addresses the needs and requirements of the LHC experiments. The current development effort is focused on further improvement of system performance, a partial redesign aimed at more complete decoupling of the user interfaces from the data repository, and the integration of new technologies such as AJAX for user interfaces.

Acknowledgements

This work was funded by EGI-InSPIRE. EGI-InSPIRE is a project funded by the European Union under contract INFSO-RI-261323.

References

[1] Evans L and Bryant P 2008 LHC machine. J. Instr. 3:S08001. doi:10.1088/1748-0221/3/08/S08001
[2] Shiers J 2006 The Worldwide LHC Computing Grid (worldwide LCG) *Proc. of the Conference on Computational Physics (CCP06) Computer Physics communications, vol. 177, pp 219–223*
[3] Legrand I C et al. 2004, MonALISA *Proc. of CHEP04, Interlaken, Switzerland*
[4] Casey J et al. 2009 Monitoring the efficiency of user jobs *Proc. of CHEP09, Prague, Czech Republic*
[5] The ATLAS Computing Group 2005 ATLAS computing technical design report. ATLAS-TDR-017, CERNLHCC-2005-022
[6] DDM Dashboard awstats, http://lxarda20.cern.ch/awstats/awstats.pl?month=09&year=2010
[7] Collados D 2009 Evolution of SAM in an enhanced model for monitoring WLCG services *Proc. of CHEP09, Prague, Czech Republic*
[8] CERN Courier 2009, http://cerncourier.com/cws/article/cern/40106
[9] http://dashb-atlas-data.cern.ch/dashboard/request.py/site?fromDate=2010-05-01%2000:00&toDate=2010-06-01%2000:00
[10] Andreeva J et al. 2010 Experiment Dashboard for Monitoring Computing Activities of the LHC Virtual Organizations *J. Grid Comp. 8 pp 323-339*
[11] Andreeva J et al. 2009 Job Monitoring on the WLCG scope. Current status and new strategy *Proc. of CHEP09, Prague, Czech Republic*
[12] Nilsson P 2008 PanDA System in the ATLAS Experiment *Proc. of XII Advanced Computing and Analysis Techniques in Physics Research (ACAT08), Erice, Italy*
[13] Letts J et al. 2010 Measuring and understanding computer resource utilization in CMS *Proc. of CHEP10, Taipei, Taiwan*
[14] Karavakis E et al. 2009 CMS Dashboard Task Monitoring: A user-centric monitoring view *Proc. of CHEP09, Prague, Czech Republic*
[15] Andreeva J et al. 2010 Visualization of the LHC Computing Activities on the WLCG Infrastructure *Proc. of CHEP10, Taipei, Taiwan*
[16] Belforte S et al. 2009 The commissioning of CMS sites: improving the site reliability *Proc. of CHEP09, Prague, Czech Republic*
[17] CMS Widgets, http://eco.netvibes.com/users/4051/gonis/widgets