Experiment Dashboard - a generic, scalable solution for monitoring of the LHC computing activities, distributed sites and services

J Andreeva\textsuperscript{1}, M Cinquilli\textsuperscript{1}, D. Dieguez\textsuperscript{1}, I Dzhunov\textsuperscript{1}, E Karavakis\textsuperscript{1}, P Karhula\textsuperscript{1}, M Kenyon \textsuperscript{1}, L Kokoszkiewicz\textsuperscript{1}, M Nowotka\textsuperscript{1}, G Ro\textsuperscript{1}, P Saiz\textsuperscript{1}, L Sargsyan\textsuperscript{2}, J Schovancova\textsuperscript{3}, D Tuckett\textsuperscript{1}

\textsuperscript{1} European Organization for Nuclear Research, CH-1211 Genève 23, Switzerland
\textsuperscript{2} A.I. Alikhanyan National Scientific Laboratory (AM)
\textsuperscript{3} Institute of Physics, Academy of Sciences of the Czech Republic, Na Slovance 2, CZ-18221 Prague 8, Czech Republic

E-mail: pablo.saiz@cern.ch

Abstract. The Experiment Dashboard system provides common solutions for monitoring job processing, data transfers and site/service usability. Over the last seven years, it proved to play a crucial role in the monitoring of the LHC computing activities, distributed sites and services. It has been one of the key elements during the commissioning of the distributed computing systems of the LHC experiments.

The first years of data taking represented a serious test for Experiment Dashboard in terms of functionality, scalability and performance. And given that the usage of the Experiment Dashboard applications has been steadily increasing over time, it can be asserted that all the objectives were fully accomplished.

1. Introduction
The Worldwide LHC Computing Grid (WLCG)\cite{1} provides storage and computational power for four LHC experiments ALICE, ATLAS, CMS and LHCb. The WLCG infrastructure is huge and heterogeneous. It includes about 150 computing sites with several different middleware flavours. The success of operations of such an infrastructure depends strongly on the monitoring of the distributed sites and services and of the quality and scale of the LHC computing activities. There are multiple monitoring systems which are used for monitoring of the WLCG infrastructure and of the LHC computing activities. Most of them were developed by the computing teams within the LHC experiments and are coupled with the particular workload management systems or data management systems. Taking a completely different approach, the Experiment Dashboard aims to provide common solutions in order to show cross-experiment and cross-technology views of the job processing, data transfer and site commissioning. The main challenge for the development of the generic monitoring systems consists of careful analysis of the similarities of the computing models and operations of various experiments and good understanding of their differences. Proper design of the generic system should allow to develop the core part which provides common functionality with a possibility to easily adapt the system for the requirements of a given experiment. Another important challenge consists of satisfying the required scalability.
and performance. The design of the monitoring system should foresee the capability to grow and scale along with the steady growth of the amount of the WLCG computing resources, of the number of users and of the scale of the LHC activities. For example, during last 12 months, the number of jobs processed by the CMS Virtual Organization (VO) almost doubled from 350K jobs to 650K jobs daily and the number continues growing.

During first years of data taking the Experiment Dashboard framework played an important role in the operations of the WLCG infrastructure and proved to provide all the necessary functionality and performance.

There are multiple Dashboard applications developed for various types of the computing activities and focused on the needs of different user categories. All of them share a common framework which is steadily evolving and adapting to new technologies.

The Experiment Dashboard [2] offers multiple applications which are shared by several LHC Virtual Organizations. The applications can be divided in four categories, as shown in Fig 1.:

- Job processing monitoring, namely the Task monitoring for analysis and production users, the Interactive view which provides the complete picture of the real-time job processing situation, and the accounting portal, so called Historical view
- Data management monitoring, namely the ATLAS Distributed Data Management Dashboard and the WLCG Transfer Dashboard. The latter was developed recently in order to provide global view of the data transfers on the WLCG infrastructure
- Infrastructure monitoring, namely Site Status Board, Site Usability Portal and SiteView. SiteView focuses on the needs of the support teams at the sites which serve several LHC VOs
- Publicity and dissemination purposes, like WLCG GoogleEarth Dashboard

Most of applications mentioned above are used by at least by two of LHC VOs: ATLAS and CMS. The usage of the Experiment Dashboard applications is steadily growing. The access to the applications is being monitored in order to understand usage pattern, to tune deployment model and to figure out possible improvements regarding system functionality and performance.
Moreover, Dashboard provides information for other applications, among them Imperial College Real Time Monitor and the CMS Popularity [3] which shows CMS data usage statistics.

The next section of this paper describes the main design principles of the Experiment Dashboard system. The following sections briefly describe the functionality and usage of the job monitoring, data transfer applications and applications for monitoring of the distributed sites and services. The final part of the paper presents the development plans and contains the conclusions.

2. Architectural and design principles

The Experiment Dashboard framework provides all the necessary components to build monitoring applications. First of all, there are collectors that gather information and store it using a Database Access Object (DAO). The DAO layer offers a decoupling between the application and the database engine. Then, a web server provides the access to read the information from the database. The web server also uses the DAO, and then it converts the data to the format requested by the user: HTML, XML, json, csv, text or plots.

Thanks to the use of external JavaScript libraries, the Experiment Dashboard can build applications that are intuitive and with all the required functionalities [4]. All the Experiment Dashboard applications have switched from server-side HTML generation to a client-side Model-View-Controller (MVC) architecture and a common cocktail of technologies, principally jQuery and selected plug-ins.

There is a build system that prepares the distribution of all the Dashboard applications. The build system keeps several versions of each application: nightly builds, unstable and stable releases. Having multiple versions allows to have several instances of the applications, one for development with the latest features, and a production one where the changes are introduced only after they have been validated on a development server by the user community.

3. Job Monitoring

Both the ATLAS and CMS Virtual Organisations use various fully distributed job submission methods and execution back-ends. More than 700,000 ATLAS and 650,000 CMS jobs are submitted daily to the Worldwide LHC Computing Grid (WLCG) and are processed on different middleware platforms. The LHC job processing activity is mainly divided into two categories: the processing of large scale Monte-Carlo production and data reconstruction jobs, and user analysis jobs. The main difference between these categories is that the former is a well-organised activity performed by a group of experts, while the latter is chaotic analysis processing by diverse and geographically widespread members of the physics community. The behaviour of analysis jobs is particularly difficult to predict as it is normally carried out by users who are not necessarily experienced in using the Grid. All of these factors increase the complexity of monitoring the job processing activities within these VOs. While most of the existing monitoring applications are coupled to a specific workload management system, the Experiment Dashboard supports different middleware implementations and job submission systems. The Experiment Dashboard monitoring system has loose coupling to the primary information sources adding flexibility to the system, thus, it combines Grid monitoring data with information that is specific to the VO by collecting information from various sources, such as the job submission systems of a VO and the jobs themselves running at the worker nodes, presenting all this information in a coherent way, as if all of it came from one single source.

The Dashboard Job Monitoring relies on the instrumentation of the job submission frameworks and, for that purpose, a common set of libraries is provided. It also defines a common set of attributes and format for the reporting of the job status updates. The job submission systems and the jobs running on the worker nodes are instrumented to report any job status updates to a messaging server that can be either MonALISA [5], used for CMS jobs, or the
Apache ActiveMQ Messaging System for the Grid (MSG) [6], used for ATLAS jobs submitted with Ganga [7]. A python agent, the Dashboard consumer, collects the job status updates and stores them in the Dashboard Data Repository which is implemented in ORACLE. In case of ATLAS, there are two more Dashboard consumers that collect information from the database of the ATLAS Production System and the PanDA Workload Management System[8] and then store and import it in the Dashboard Data Repository. In order to ensure the reliability of the service, the Dashboard consumers are being monitored on multiple levels:

- On the database level with database alarms
- On the run-time level by checking for error conditions
- On the server level with Linux cron job scripts

All the collected information is then presented to the user via the User Web Interfaces and the Dashboard web servers. The same data repository is used by multiple applications (User Web Interfaces) within a VO. Each of them is focused on a particular use case and targets a specific category of users. The architecture is shared between ATLAS [9] and CMS [10] even though they are using different job submission systems and execution back-ends.

The User Web Interfaces are not only VO agnostic but they are also database agnostic, thus, it makes no difference if the data back-end is implemented in Oracle, MySQL, any NoSQL solution or even a simple file on the users hard disk, as long as the information that is being passed to the User Web Interface is in the correct format. The web front-ends, both for the ATLAS and CMS collaboration, expose modern user-interfaces built with hBrowse [11], a common jQuery framework used for generic job monitoring applications. The User Web Interfaces for the monitoring of the job processing activity are divided into three parts:

- a) generic job monitoring
- b) monitoring of the user’s analysis and production activities
- c) providing historical data for long-term statistics.

The information stored in the Dashboard Data Repository is also consumed by other applications, either internal Dashboard applications such as the WLCG Google Earth Dashboard, or external third party ones such as the Imperial College Real-Time Monitor system, in machine-readable format through a set of well-defined APIs.

3.1. 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 the jobs submitted or terminated in a chosen time period categorised by their activity such as analysis, production and testing jobs. The application provides a wide flexibility to the users. They can filter and sort all the jobs by quite a lot of different properties, such as the execution site, the grid gateway, the user, the completion status, the execution host, the activity type and the dataset used. The users can also navigate from an overview with the number of jobs on each state to a page with very detailed information about a particular job, like the exit code and exit reason, important time stamps of processing the job and the number of processed events.

3.2. Monitoring of the user’s analysis and production activities
The Analysis and Production Task Monitoring applications expose a user-centric set of information to the user regarding submitted tasks. They provide a clear and precise view of the status of the jobs submitted to the Grid with very low latency. The status of the 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. They play a very important role in the distributed analysis and production operations of both ATLAS and CMS as they empower the users in such a way so that only serious issues are escalated to the user support teams. The users can easily detect sites with low job processing efficiencies or sites that are in maintenance so that they can choose to cancel their jobs and resubmit them to a different site. In addition, they help support teams to solve user problems by offering advanced job and task debugging information.

The Analysis and Production Task Monitoring applications were proven to be an essential component of LHC computing operations; more than three hundred and fifty unique users are using them daily for their work just for CMS. The Analysis Task Monitoring application has been recently ported to the Android platform and it now allows users to follow the progress of their tasks from their Android smart-phone or tablet.

3.3. Accessing historical data for long-term statistics
The Historical Views 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 advanced graphical 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. It provides a wide selection of graphical plots for selected parameters and helps to understand the nature of the infrastructure inefficiencies, reason of failures, and assists in resolving and predicting problems. The Historical Views application is widely used by site admins, and VO computing managers.

4. Transfer monitoring
Distributed data management at LHC scales is a complicated task. It deals with challenging management issues related to heterogeneous storage systems and wide area networks. Reliable monitoring is therefore extremely important for the organization and the operation of the LHC data management. Initially the Experiment Dashboard system provided a solution which was developed for a single LHC VO, ATLAS. ATLAS DDM Dashboard is one of the core components of the ATLAS Monitoring infrastructure. It is widely used by the ATLAS computing community. Usage of the application increased twice from the beginning of 2012. ATLAS DDM Dashboard is accessed daily by about 50 unique visitors.

The data management monitoring systems currently used on the WLCG infrastructure were created within a particular LHC VO and work in the scope of a single experiment. There was no global cross-VO view of ongoing transfers performed on the WLCG infrastructure. The WLCG Transfer Dashboard[12] which was developed recently aimed to solve this problem. The goal of the system is to provide cross-VO and cross-technology view of the WLCG data transfers. For performing data transfers, three LHC experiments ATLAS, CMS and LHCb use the File Transfer Service (FTS)[13]. ALICE VO mainly relies on xRootD[14] both for data transfer and data access. Therefore the implementation of the global data transfer monitoring system started from monitoring of the data transfers handled by FTS. The next phase is enabling of monitoring of the xRootD data transfers.

The WLCG Dashboard Transfers was developed in the Experiment Dashboard framework following the design principles described in the second section of this paper. All distributed FTS services were instrumented for reporting monitoring information via MSG. Dashboard collector consumes information from MSG and records it into the central repository implemented in ORACLE 11g. Transfer statistics are regularly generated with minimum time bin of 10 minutes. The monitoring data is exposed through the UI. All potential clients, for example monitoring systems of the LHC VOs can use provided APIs in order to retrieve data in machine
readable format (XML, JSON). In the development of the WLCG Transfer Dashboard, all the experience accumulated during the development of the ATLAS DDM Dashboard and operational data transfer experience in general were taken into account. The new system re-used several components of the ATLAS DDM Dashboard, for example the schema of the repository, the statistics aggregation and the user interface. This allowed to develop a first prototype in a few months. Then, a thorough consistency checks between the WLCG Transfer Dashboard and the VO-specific monitoring systems PhEDEx and ATLAS DDM Dashboard were performed. All the discrepancies detected during the first round of the consistency checks were understood. All FTS servers were patched in order to fix the problem with the ActiveMQ-cpp client. Second round of the consistency checks showed very good agreement between WLCG Transfer Dashboard and PhEDEx and ATLAS DDM Dashboard.

As mentioned above, the next development steps should allow to integrate xRootD transfers in the global transfer monitoring system. The native xRootD monitoring is performed with UDP. It is split in two monitoring flows, the summary flow which contains aggregated data transfer and data access metrics and the detailed flow. In order to provide information which is required for the WLCG Transfer Dashboard, the detailed flow should be processed in order to generate event-like monitoring reports which contain file level transfer information including time stamps when transfer started and finished, the names of the files, the IP address of the source and destination, number of transferred bytes. These reports will be published to MSG from where they can be consumed by the Dashboard collector. The new version of the WLCG Transfer Dashboard, including the xRootD transfers, should be deployed in production by the end of 2012.

5. Site and service monitoring
There are three Experiment Dashboard applications focused on the status of sites and services. All of them are described in the following sections.

5.1. Site Status Board
The Site Status Board (SSB) is a tool to find the answer to a simple question ‘Is site X working properly?’. Even if the question is very straight forward, the answer can be very difficult. It might depend on quite a lot of different parameters, and those parameters are usually different for each experiment.

With this idea in mind, the SSB provides a framework where the responsible for the experiments can define their own metrics. A metric can be anything that can be monitored: number of jobs executed or queued at a site, status of different services, software installed at the site, downtime information, etc. Metrics can have a numerical value, a text value, a color and a link to more information. The color represents the status of the metric: green if everything is fine, yellow in case of warning, red for problems and brown for maintenance. SSB will collect these metrics over time and present them to the users.

On top of that, metrics can be combined into views. A view presents a set of metrics, and it calculates for each site a single icon showing the combined status of all the metrics: downtime if one metric is brown, a red ball if at least one metric is red, or a nice green tick if all the metrics are green. Then, it is very easy to spot the sites currently having problems.

One of the strongest points of the SSB is that the flexibility to add new metrics and create or modify views. Thanks to this, the responsible for the experiment can try out different metrics and see which ones are the important ones.

Like all the other applications presented on this paper, the SSB is built using the Experiment Dashboard framework. The structure is also similar to the other applications: there are several collectors that gather data and insert it on an ORACLE database, and a web server that presents
this information in multiple formats: html pages and plots for users, and also machine readable format like JSON or XML.

There are three categories of people in the SSB world:

- The SSB developers, who provide the framework.
- The responsible for the experiments. They can authenticate to the SSB using their certificates, and then they can use the SSB web pages to add or modify metrics or views.
- The end user, who looks at the views setup by the responsible, and can monitor the status of the sites.

The SSB was developed following a request from one experiment, CMS. Since the very beginning, the SSB was designed in such a way that it could be used by several communities. And this extra effort paid off. The SSB is currently setup for the four main LHC experiments, and CMS and ATLAS heavily rely on it for their computing shifts and site commissioning exercises. The CMS responsibilities have defined more than a hundred metrics, and combined them into ten different views. The CMS SSB is visited weekly by more than two hundred unique visitors from 27 countries. The ATLAS responsibilities have defined more than two hundred metrics and thirty views. The ATLAS SSB is visited weekly by more than one hundred-thirty unique visitors.

5.2. Siteview

Siteview is an application very similar to the Site Status Board. The main difference is the target audience: whereas SSB is used within an experiment to monitor all the sites that provide resources to that experiments, Siteview is intended for site administrators, and it shows for each site the status of the LHC experiments that run on that particular site. Thanks to this application, site administrator can check on a single page if all the experiments running correctly.

SSB and Siteview have the same database layout, and the collector structure, thus simplifying the development and maintenance of both applications. In the case of Siteview, the Experiment Dashboard group is also in charge of defining the metrics and views.

The web interface is quite different. Since Siteview is site-centric, the main page presents a site and four areas per experiment running on the site:

- The first area with the overall status of the Experiment at the site
- The second one presents the status of jobs: running and pending, and grouped by the type of jobs
- Then, the transfers into the site
- Finally, the transfers from the site to the rest of the world

5.3. Site Usability Monitoring

The WLCG Collaboration consists of more than 150 Grid computing centres distributed world-wide connected through network with either dedicated or public bandwidth. These computing centres host computing resources such as CPU resources capable of running over 200k computational jobs simultaneously, hundreds of PB of data storages on disks and tapes. The LHC experiments benefit from the infrastructure provided by the WLCG project. Given the magnitude and distributed nature of the computing resources, their monitoring is a challenge.

The SAM/Nagios framework [15], [16] is used to perform sane testing health of the critical resources provided by the sites, such as availability of computing elements and storage systems. Availability and reliability of site services and of sites based on the test results is then used in reports to the WLCG Management Board.

The Experiment Dashboard framework provides the Site Usability Monitoring (SUM) to visualise the test results, and the calculated availabilities and reliabilities of the grid sites and of
the services hosted at the sites. The main four LHC experiments, ALICE, ATLAS, CMS, and LHCb benefit from the SUM User interface.

The SUM UI provides two types of views: the Latest results view and the History view. The Latest results view shows availability or reliability of all service endpoints in the last 12 hours. The Latest results view, as well as the History view offers possibility to filter sites, service flavours, metrics, and test exit codes.

The History view visualises availability or reliability of a site service or a site either as a heat map quality plot, or as a ranking plot. The History view offers granularity not only in entity selection, time interval, but also in binning. The SUM UI user can start with the view of the site availability, click through to the availability of services at that site, drilling down to overview of the test results for 1 particular endpoint, ending with log file of a particular test. Fig 2. presents an example of a quality plot and Fig 3. a ranking plot of monthly reliability.

The color scheme of the history plot allows the SUM UI user to identify issues with a site or a service. The time binning allows the SUM UI users to determine the issue duration retrospectively in time. The Site Usability Monitoring targets at the computing operations experts of the LHC experiments, shifters, and the site administrators.

6. Future work
The plans for the Dashboard development are aligned with the requirements of the LHC experiments. New versions of the Task monitoring for production and analysis users were currently deployed in production for the ATLAS community. A lot of new feature requests were received from the ATLAS production team regarding production Task monitoring. For the analysis users new version of the task monitoring should allow not only to visualize status of the analysis tasks but also to kill and to resubmit jobs using analysis task monitoring UI.

The prototype of the new accounting portal for space storage was recently deployed for validation by ATLAS. This application is currently in the active development phase.

The main development effort is given to the applications which are shared by all LHC VOs. Among them is WLCG Transfer Dashboard. Along with enabling of the xRootD transfers in the new global monitoring system there is a long list of feature requests which was collected from the LHC computing community. Current version of the WLCG Transfer Dashboard provides basic functionality which allows to monitor transfer throughput and to detect transfer failures. The
LHC VOs required functionality which would allow to track and to understand various transfer latencies and inefficiencies.

Following the outcome of the WLCG Operations TEG further development is foreseen for the SiteView applications. It should facilitate the task of the support teams at the distributed sites supporting several LHC VOs.

7. Conclusions
The Experiment Dashboard proved to be an essential component of the WLCG monitoring infrastructure. It is widely used by the LHC experiments, in particular by ATLAS and CMS. The Experiment Dashboard provides generic solutions which work across different VOs in the same time flexible enough to be adopted for requests and use-cases of a particular VO. The proper design allows to easily extend the applications initially developed for a single experiment to the global scope, for example recently developed WLCG Transfer Dashboard shares a lot of implementation with the ATLAS DDM Dashboard. The Dashboard applications serve various categories of the LHC users, among them the WLCG and VO management, production teams, site administrators, LHC physicists running their analysis tasks on the GRID. The usage of the Experiment Dashboard is steadily growing, in terms of number of requests, volume of stored and retrieved data, number of unique visitors. Along with the constant growth of the WLCG infrastructure and of the scale of the LHC computing activities the Experiment Dashboard is evolving fast providing needed scalability and performance and adapting new technologies. Further development of the Experiment Dashboard will follow the needs of the LHC computing community.

References
[1] Worldwide LHC Computing Grid http://wlcg.web.cern.ch/
[2] J. Andreeva et al, Experiment Dashboard for monitoring of the LHC distributed computing systems, 2011, J. Phys.: Conf. Ser. 331 072001 doi:10.1088/1742-6596/331/7/072001
[3] D. Giordano et al., Implementing data placement strategies for the CMS experiment based on a popularity mode, 2012, to appear in the proceedings of Computing in High Energy and Nuclear Physics (CHEP12)
[4] J. Andreeva et al., Designing and developing portable large-scale JavaScript web applications within the Experiment Dashboard framework, to appear in Proceedings of the Computing in High Energy and Nuclear Physics 2012 International Conference
[5] I. Legrand et al, MonALISA: An agent based, dynamic service system to monitor, control and optimize distributed systems, 2009, Computer Physics Communications 180, Issue 12 doi:10.1016/j.cpc.2009.08.003

[6] J. Casey et al, Monitoring the efficiency of user jobs, 2010 J. Phys.: Conf. Ser. 219 072017 doi:10.1088/1742-6596/219/7/072017

[7] M. Kenyon et al, Key developments of the Ganga task-management framework, 2012, to appear in the proceedings of Computing in High Energy and Nuclear Physics (CHEP12)

[8] T. Maeno, PanDA: Distributed production and distributed analysis system for ATLAS, J. Phys. Conf. Ser. 119 (2008) 062036

[9] J. Andreeva et al, ATLAS job monitoring in the Dashboard Framework, 2012, to appear in the proceedings of Computing in High Energy and Nuclear Physics (CHEP12)

[10] E. Karavakis et al, CMS Dashboard Task Monitoring: a user-centric monitoring view, 2010, J. Phys.: Conf. Ser. 219 072038 doi:10.1088/1742-6596/219/7/072038

[11] L. Kokoszkiewicz et al, hBrowse - Generic framework for hierarchical data visualization, in proceedings of EGI Community Forum 2012 / EMI Second Technical Conference, 2012, PoS(EGICF12-EMITC2)062

[12] J. Andreeva et al., Providing global WLCG transfer monitoring; to appear in Proceedings of the Computing in High Energy and Nuclear Physics 2012 International Conference

[13] A. Frohner et al., Data management in EGEE, Journal of Physics Conference Series 219 062012, doi: 10.1088/1742-6596/219/6/062012, 2009.

[14] The Scala Software Suite: xrootd/cmsd, http://xrootd.slac.stanford.edu/

[15] P. M. Rodrigues De Sousa Andrade et al, Service Availability Monitoring framework based on commodity software, to appear in Proceedings of the Computing in High Energy and Nuclear Physics 2012 International Conference

[16] A. Di Girolamo et al., New solutions for large scale functional tests in the WLCG infrastructure with SAM/Nagios: the experiments experience, to appear in Proceedings of the Computing in High Energy and Nuclear Physics 2012 International Conference