Dashboard for the LHC experiments.

J. Andreeva[1], S. Belov[2], A. Berejnoj[3], C. Cirstoiu[1,4], Y.Chen[5], T.Chen[5], S. Chiu[5], M. De Francisco De Miguel[1], A. Ivanchenko[1], B. Gaidioz[1], J. Herrala[1], M. Janulis[1], O. Kodolova[3], G. Maier[1], E.J. Maguire[1], C. Munro[6], R. Pezoa Rivera[1], R. Rocha[1], P. Saiz[1], I. Sidorova[1,2], F.Tsai[5], E. Tikhonenko[3], E. Urbah[7]

[1] European Organization for Nuclear Research CERN G06910, CH-1211 Geneva 23, Switzerland
[2] Joint Institute for Nuclear Research, 141980, Dubna, Moscow Region, Russia,
[3] Moscow State University, 119899, Moscow, Russia,
[4] Politehnica University in Bucharest, 060042, Bucharest, Romania,
[5] Academia Sinica, Taipei 115, Taiwan,
[6] Brunel University, Uxbridge, London UB8 3PH, United Kingdom
[7] Laboratoire de l'Accelerateur Lineare, Universite Paris-Sud 11, IN2P3/CNRS, Paris, France

E-mail: Julia.Andreeva@cern.ch

Abstract. In this paper we present the Experiment Dashboard monitoring system, which is currently in use by four Large Hadron Collider (LHC)[1] experiments. The goal of the Experiment Dashboard is to monitor the activities of the LHC experiments on the distributed infrastructure, providing monitoring data from the virtual organization (VO) and user perspectives. The LHC experiments are using various Grid infrastructures (LCG[2]/EGEE[3], OSG[4], NDGF[5]) with correspondingly various middleware flavors and job submission methods. Providing a uniform and complete view of various activities like job processing, data movement and publishing, access to distributed databases regardless of the underlying Grid flavor is the challenging task. In this paper we will describe the Experiment Dashboard concept, its framework and main monitoring applications.

1. Introduction

The LHC experiments ALICE, ATLAS, CMS and LHCb are preparing for data acquisition planned to start in 2008. The LHC will produce about 15 Petabytes of data annually and access to this data has to be provided to 5000 scientists in about 500 scientific institutions all over the world. These numbers allow to estimate the scale and complexity of the LHC computing task. For their computing systems all LHC experiments had chosen the globally distributed tiered model. The LHC experiments are relying on several Grid infrastructures: LCG/EGEE, OSG, NDGF. These infrastructures have been built and deployed to the scientific communities over the last years. At the current stage the improvement of the performance and reliability of the Grid sites and services is becoming a very important task and various monitoring systems are contributing to achieve this goal. At the same time the overall operation depends not only on the Grid's performance and reliability, but also on the quality of the experiment specific software and services. The monitoring system focused on the needs

of the LHC user community has to provide monitoring information in a transparent way across several Grid infrastructures and has to combine data related to the performance of the Grid infrastructure and data that is experiment/activity/application specific. Such a monitoring system should be able to satisfy users in various roles including production manager, coordinator of data transfer, site administrator supporting a given experiment at the local site or simply an analysis user submitting analysis jobs to the Grid. Finally, it should be capable not only to detect different kinds of problems or inefficiencies, but also to assist in understanding the underlying reasons.

The goal of the Experiment Dashboard is to satisfy the requirements mentioned above and in particular satisfy the needs of the given experiment to follow the wide spectrum of its activities such as job processing, data transfer, data access and access to the distributed data bases. Taking into account the scale of the used infrastructures (the largest production Grid infrastructures available today) and the number of users of the virtual organizations, the performance and scalability of the system is an important issue which has to be addressed.

In the first section of this paper we will describe the concept of the Experiment Dashboard monitoring system and its framework. The next sections will provide an overview of the main Experiment Dashboard monitoring applications, including examples of the usage of the tools by the LHC experiments. The two final sections will focus on the future work and draw some conclusions.

2. Overview of the architecture

The Experiment Dashboard provides a view of the Grid infrastructure focused on the needs of the users of the LHC VOs. Conceptually the development was not focused on the creation and deployment of the new sensors. Instead, the dashboard developers took into account the requirements of the LHC experiments and analysed monitoring data collected by other monitoring systems and by the experiment specific tools and services. This data was analysed in order to understand how various sources can complement each other and how information coming from multiple sources can be correlated and merged. In case when required data was not available in the existing systems, Dashboard developers worked together with the developers of the LHC experiments to define modifications required in the workload management or data management systems for reporting missing information to the Experiment Dashboard.

The main components of the Experiment Dashboard are information collectors, data storage, currently implemented for an Oracle backend, and the services responsible for data retrieval and information presentation. Depending on the use case and provided functionality, the Experiment Dashboard is using multiple sources of information, among them other Grid monitoring systems like R-GMA (Relational Grid Monitoring Architecture)[6], GridIce (Monitoring tool for Grid Systems)[7], SAM (Service Availability Monitoring)[8] or IMRTM (Imperial College Real Time Monitoring of the Resource Brokers)[9], experiment specific services running at the local sites like the ATLAS Data Management services, experiment central databases like the ATLAS Production database and servers of the MonALISA (Monitoring Agents using a Large Integrated Services Architecture)[10] monitoring system (described in the following chapters). Information can be transported to the Experiment Dashboard via various protocols and data collectors are implemented in either push or pull modes.

2.1. Introduction to the Experiment Dashboard Framework

All monitoring applications regardless of the actual implementation of the data collectors, data transfer protocols or formats of the input information are developed within the Experiment Dashboard framework. The schema of the Experiment Dashboard framework is presented in figure 1. The framework is implemented in Python and covers the main functionalities of all monitoring applications: collection of information, access to the database and presentation of the information to the user, using the Apache HTTP server and its mod_python extension (Apache module that embeds the Python interpreter within the server).

2.1.1. *Information collectors and other Dashboard agents*. The collection of input information implies regular access to the information source for retrieval of the monitoring data and storage in the Experiment Dashboard database. To provide a reliable monitoring system, data collectors should run permanently and need to recover any missing data in case of eventual failure and following restart. The Experiment Dashboard framework provides all the necessary tools to manage and monitor these agents, each focusing on a specific subset of the required tasks. The status of the collector, as well as error messages it publishes, are automatically reported to the Experiment Dashboard maintainers. The system supports three reporting methods: a web application showing the status of all services, e-mail or SMS. This automatic monitoring feature is very useful for all kinds of permanent tasks (tasks which should be performed on regular basis), not only collection of information. For example, they are used to implement computations of daily statistical summaries or analyze the collected data and generate alarms.

2.1.2. *Database access*. To ensure a clear design and maintainability of the system, the definition of the actual monitoring application queries is decoupled from the internal implementation of the data storage. Every monitoring application implemented within the Experiment Dashboard framework comes with the implementation of one or more Data Access Objects (DAO), which represent the "data access interface": a public set of methods for the update and retrieval of information. Access to the database is accomplished by using a connection pool to reduce the overhead in creating new connections, therefore the load on the server is reduced and the performance increased.

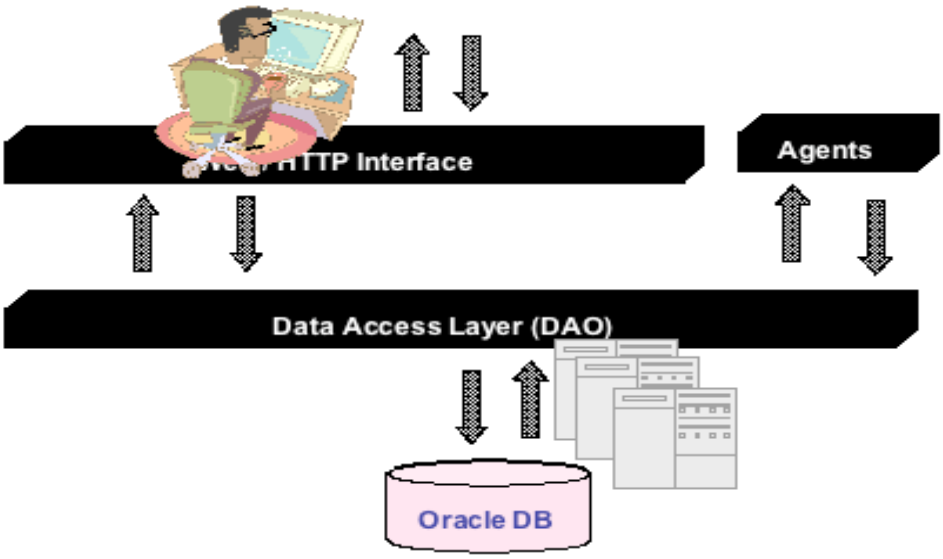

Figure 1. Experiment Dashboard Framework

2.1.3. *Data presentation layer*. The Experiment Dashboard requests are handled by a system following the Model-View-Controller (MVC) pattern. They are handled by the "controller" component, launched by the mod_python extension, which keeps the association between the requested URLs and the corresponding "actions", executing them and returning the data in a format requested by the client. All actions process the request parameters and execute a set of operations, which may involve accessing the database via the DAO layer. When a response is expected, the action

stores it in a python object, which is then transformed into the required format (HTML page, plain XML, CSV, image) by the "view" components. The controller applies automatically the view to the data.

A widely used format for data retrieval is HTML, so that data can be viewed in any browser, but information can also be retrieved in XML (eXtensible Markup Language), CSV (Comma Separated Values) or image formats. This flexibility allows the system to be used not only by users but also by other applications. A set of command line tools is also available. Most of the time clients request information in common formats like XML or CSV. For these cases the framework does the necessary transformation automatically, without the need for a specific view being developed. In other cases the output format is not so generic and an extension to this class has to be implemented.

The HTML pages are in general specific to the information they contain, since they can include various interactive functionalities for data browsing. The Experiment Dashboard framework doesn't provide a generic HTML rendering of the output data, however, it applies an XSL transformation to the XML representation of the information retrieved from the database. In way the system generates an HTML page from the output data.

3. Experiment Dashboard applications

Experiment Dashboard monitoring applications are covering a wide range of activities of the LHC experiments on the Grid. Some of the applications, like job monitoring and site reliability, are generic and in production for all LHC experiments, others are specific to a given experiment, like data management monitoring [11], developed on the request of the ATLAS experiment. As a general rule, the development of a new application starts on the request of one of the LHC experiments, but in case when no experiment specific information sources are involved, this application can be reused for other experiments. The overview of currently available applications is shown in figure 2. The main applications are described in this chapter.

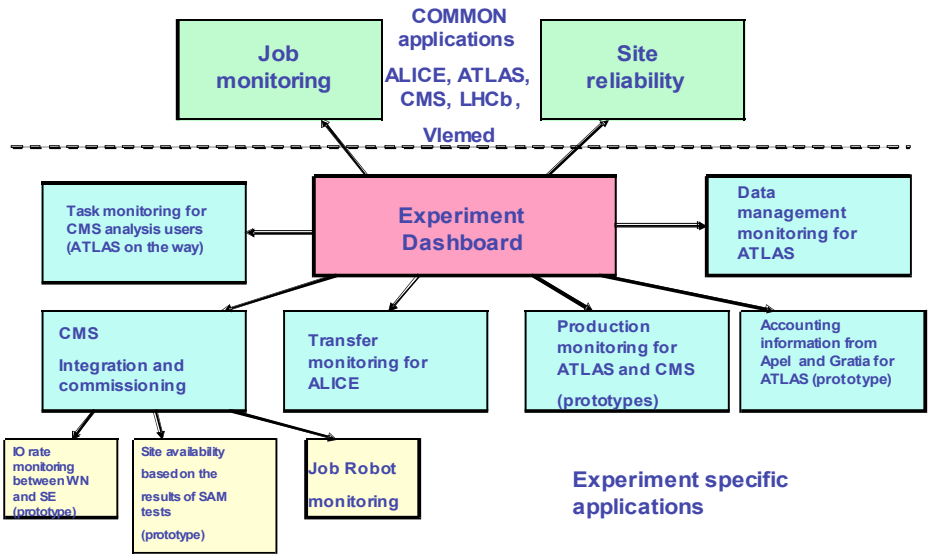

Figure 2. Experiment Dashboard applications

3.1 Job monitoring

The goal of the job monitoring is to follow job processing of the LHC experiment on the distributed infrastructure. The Experiment Dashboard keeps track of all jobs submitted by the experiment users

and stores the main monitoring indicators, such as resource usage and sharing, Grid behavior, application robustness and data access quality. Information related to the job processing is aggregated and depending on the users's request can be shown as a function of different attributes, like user, activity jobs belong to, site, computing element where jobs are processed, Resource Broker, application, input collection, job status, or result of a job's processing. The Experiment Dashboard shows relevant quantities of the current state of the experiment activities on the Grid, for example, how many jobs are running, pending or completed, and which fraction of the completed jobs was successful. The main quantities in resource utilization are CPU, memory consumption and input-output rates. The Experiment Dashboard also helps to keep record of resource sharing between physics activities within an experiment, such as production and analysis, between different analysis groups and between individual users.

For job monitoring the Experiment Dashboard uses multiple sources of information. In order to compile a comprehensive picture of the overall job success rate, the data collectors gather both Grid-related information and information specific to the application which is run by the users. The Grid-related information is obtained from the Information System (active sites and queues) and from the Logging and Bookkeeping system of the Resource Brokers (job status changes and destination queues for individual jobs). Information from the Logging and Bookkeeping system is not retrieved directly. The Experiment Dashboard uses RGMA and ICRTM. The application-specific information is gathered throughout the job lifetime – submission, runtime and output retrieval – via the MonALISA monitoring system, developed by Caltech University. In order to communicate job status information and application- specific information to the Experiment Dashboard, the job submission tools of the ATLAS and CMS experiments and the job wrappers generated by these tools are instrumented to report meta information about user tasks and the progress of the user job at the worker node to the MonALISA server. The Experiment Dashboard collector retrieves data from the MonALISA server via the web service interface. Information related to a single job and coming from different sources is correlated via the Grid job identifier. Due to the use of the MonALISA system, the Experiment Dashboard provides job monitoring functionality regardless of the job submission method or the middleware flavor and works across various Grid infrastructures. The possibility to provide a complete and transparent view for job processing for multiple Grid infrastructures is vital for experiments like ATLAS and CMS, which are heavily using several infrastructures and different job submission methods. For example, during the Computing Software and Analysis Challenge run by the CMS experiment in 2006, the Experiment Dashboard was the only monitoring tool capable of following the full scale job processing both at the OSG and LCG infrastructures and able to present the complete runtime picture of the challenge.

Regarding the user interface, job monitoring applications provide a snapshot of the job processing status, the so called "interactive view", which allows to see what is happening now and provides the straightforward way to detect any inefficiencies related to job processing, like the misconfiguration of Grid services at sites or corrupted data. A historical view, which shows job processing evolution over any given time range, is also provided.

Currently, the job monitoring application is used by all 4 LHC experiments and by Vlemed VO[12] outside the LHC community. Vlemed is the Medical Diagnosis and Imaging subprogram of the Virtual Laboratory for e-Sciences Project.

3.2 Site reliability

Reliable and stable functioning of all Grid services at the sites is one of the main conditions for the improvement of the quality of the Grid infrastructure. The site reliability application [13] was developed to facilitate the task of the site administrators, to estimate the site performance regarding job processing and data transfer and to detect and understand eventual problems at the site. Site reliability related to job processing uses the same information sources as the job monitoring application, but allows to analyze what happened to every individual job submitted via a Resource Broker in more detail and presents the results of this analysis in different formats. One of the main

features of the site reliability application is that it estimates the job processing success rate by taking into account the success of every individual job submission attempt. Every individual job can be resubmitted by a Resource Broker several times. The site reliability application reconstructs the complete workflow the job is passing through, represents it in a graphical form and calculates the success rate based on the number of all job submission attempts. This application generates daily reports for the number of successful or failed attempts for all sites, serving a given VO and provides a ranking of the sites based on these reports. The application also generates a list of the Grid errors related to job processing by a Resource Broker with links to the corresponding documentation or troubleshooting recipes. It allows to follow the site performance evolution over the given period of time.

3.3 Task monitoring

The task monitoring application can be regarded as a supplement of the job monitoring application, but focused on the needs of the LHC analysis users submitting their jobs to the Grid. The task monitoring application provides a consistent way of following a user's analysis tasks regardless of the job submission tool, which was used for the job generation and submission, under the condition that the job submission tool was instrumented for the Experiment Dashboard reporting. Using the provided web interface, the physicist can browse the analysis tasks, which have been submitted over the chosen period of time, see meta attributes of the task or of the individual job, check the status of all jobs belonging to a given task in real time and check the reasons for failures of the individual jobs without the necessity of opening the log files. The application provides a plot, showing the distribution of the jobs of a task or a subset of jobs of a task as a function of the site, where the jobs are being processed or as a function of the Resource Broker. These distributions can help users to detect any problems related to the site or Resource Broker and to exclude the site or Resource Broker during the following resubmissions. Currently, the task monitoring application is in production for the CMS users and shortly will become available for the ATLAS user community.

3.4 ATLAS Data Management monitoring

Data distribution is one of the most challenging tasks for the LHC experiments, in particular due to very strong requirements regarding data safety and large-scale data replication. Practical management issues with the storage systems and wide-area networks increase the complexity of the data management task. The LHC experiments need reliable monitoring, capable of following the large scale distribution of the experiment datasets as well as the transfer of individual files. Since the ATLAS data management monitoring application uses ATLAS specific data management services at the sites, this application is currently considered to be ATLAS-specific. However, regarding the user interface and the overall structure of the monitoring data, there are a lot of similar features that could be applied to all LHC experiments. This might make it possible to partially reuse this application for other experiments, if required.

3.4.1 *Introduction to the Distributed Data Management of the ATLAS experiment*. The ATLAS Distributed Data Management (DDM) system has different components interacting to perform the data bookkeeping and placement tasks. Central catalogs take care of the first part – bookkeeping – while local site services try to answer user requests to spread and replicate data among the different computing locations. 11 major computing centers are involved, with tens of smaller ones also being part of the system.

Files are grouped into datasets – according to different policies – and the latter are used as single units of transfer. Requests for the availability of datasets on a site are typically made by production coordinators following a concrete plan for data distribution or from users wishing to perform analysis over the data.

The Experiment Dashboard DDM monitoring implementation took over an initial prototype done by the DDM team, which presented scalability problems. This new service has been successfully used

in production starting from the beginning of 2007. A new version, covering production data flow and transfers on the individual user's request as well as the Tier0 export, is currently under validation.

3.4.2 *Information sources*. The main sources of the monitoring information for ATLAS data management applications are ATLAS DDM site services. They provide transfer and file placement information. To understand the reasons of data management problems, the application uses the results of SAM tests and correlates results of these tests with the eventual failures detected by the monitoring system. Since the availability of the storage space is an important factor of the data management activity, the application currently uses BDII as an information source for this kind of data.

3.4.3 *Monitoring transfers*. The task of the Experiment Dashboard is to minimize the effort of system users, while tracking the evolution of data distribution. Having it done in a centralized service eliminates the need to actually contact the services running at the site to understand the problem, which is especially relevant as sites run different flavors of the local Grid fabric services - storage, local file catalogs, etc.

The flow of monitoring data starts as soon as a new dataset request - a subscription - is made and picked up by the system agents on the sites, marking it as queued. It is then marked as incomplete as soon as the first file starts to be transferred, and will eventually get to the status complete, once all the dataset files are available on the destination site. A dataset subscription can also be canceled at any time. A similar flow exists for each individual file transfer, from the moment the site transfer agents try to find a suitable replica to be used as a source - an unknown source - to the registration of the file in the local file catalog, so that end users can retrieve it - file done. Meanwhile, there are interactions with file transfer services with possible retrials in case of failure. The Experiment Dashboard keeps track of every single event in both the dataset and the file workflows, storing also the errors or reasons of every state change. This information is kept during (at least) the time of the whole dataset transfer, possibly being deleted once it is made available at the destination, and is presented in global or detailed views. The global view provides a site overview covering different metrics, for example, the throughput or a list of completed files/datasets, a summary of the most common errors (transfer and placement) , etc. Via a detailed view, the user can navigate through the complete history of the dataset transfer, starting from the dataset state to the state of each of its files and to the history of each single file placement following all status changes.

3.4.4 *Monitoring the system.* The final ATLAS use case comes from the maintainers of the system. Being highly distributed, it is very difficult to have a global picture of how the system is performing, or to understand the bottlenecks. Using the received information regarding data movement, several Dashboard tools generate a summary information for a single site or groups of sites. It includes the current measured throughput, the number of files and datasets successfully moved to the sites, a categorization of the most common transfer errors and the summary of the state of the different datasets and files on the site. The two last items are particularly useful to quickly detect and act when services on a site start to degrade. Alarms are sent whenever a given measurement goes below a defined threshold.

3.5 Monitoring of the production activity for ATLAS [14] and CMS [15].
Production activities of all LHC experiments are fully relying on the distributed Grid infrastructure. Even though there are a lot of similar features in the organization of the production activity in the LHC experiments, there are a lot of differences as well. These differences have an impact on the implementation of the production monitoring systems. Both ATLAS and CMS are using several Grid infrastructures and a variety of job submission methods for production. The functionality which is provided by the Experiment Dashboard for the production monitoring of ATLAS and CMS is different, but for both experiments the production monitoring systems are developed as a joint effort between the Experiment Dashboard team and the developers of the production systems in ATLAS and

CMS experiments. In the case of CMS, the new production monitoring system uses the Experiment Dashboard database as an information source. The Experiment Dashboard database plays the role of the central repository of the production monitoring data, but the user interface is developed by the CMS production development team. The source of input information for ATLAS production monitoring is the ATLAS production database. The ATLAS production monitoring is described in more detail below.

3.5.1 *ATLAS production monitoring*. The ATLAS production system is based on a central database holding information about the job status (waiting submission, scheduled in a resource, running, being aborted and many other). In particular, the production system manages job definitions which are then dispatched to different processing back-ends: LCG/EGEE, OSG and NDGF. The main users of the production monitoring system are production managers and operators, and the user interface should allow them to follow the production progress and to detect problems in a straightforward way.

The Experiment Dashboard production monitoring user interface is taking over from the current PHP-based implementation. The current version becomes difficult to maintain and the ATLAS experiment has requested a re-implementation of the user interface in the Experiment Dashboard framework, preserving all available functionalities, but providing the "look and feel" similar to the ATLAS data management monitoring. Because of the Experiment Dashboard framework, which provides a lot of standard methods and facilitates the development task, the new prototype was developed in two months. Currently, both user interfaces are running in parallel. This permits to check data consistency. The definite switch to the new interface will happen by the end of 2007. The re-implementation was a good opportunity to factorize multiple functionalities at once, identifying critical views, and optimizing the navigation. Due to the valuable suggestions from the ATLAS community and in particular of the ATLAS production team, the new interface has several advantages compared to the previous version. The navigation has considerably improved due to the decrease of the number of views from 10 to 3, and the increase of interactivity and the variety of filtering possibilities. The provided Python API allows to query the production monitoring data from other applications. One of the most important motivations for the development of the new interface was to better satisfy the needs of the main users of the system: production operators, production managers and site administrators. The major improvement for the production operators coming along with the new interface, is the direct way to identify the problematic tasks, which require to have action taken, to detect sites with wrong software installations (in order to send a GGUS ticket) and to get the top errors for each site/task/submission tool. Production managers and site administrators can benefit from the very flexible interface for generating performance plots which is provided by the new interface. The future development foresees the interaction between data management and production monitoring systems with the possibility of data exchange between two database back-ends. Another important direction of future development is defining alarm conditions and enabling alarms in case of problems.

3.6 Dashboard monitoring applications for the CMS commissioning and integration

One of the main customers of the Experiment Dashboard monitoring in the CMS experiment is the computing commissioning and integration. The interactive interface of the job monitoring application provides a straightforward way to detect problems related to data access, software distribution, misconfiguration of the worker nodes at sites or problems with the major Grid services like the Resource Broker. This functionality is very important for the commissoning and integration team. In addition to the job monitoring application, there were several applications developed on request of the CMS experiment, like monitoring of the input-output rate between the worker node and the storage element, the site availability view for the analysis users based on the information in BDII, and the CMS site availability based on the results of SAM tests. The CMS site availability application based on the results of SAM tests is described below in more detail.

3.6.1 *Visualization of SAM results in the Experiment Dashboard*. The LHC experiments are widely using the SAM framework for testing generic Grid services or experiment specific services at the sites as well as checking if a site is providing required functionality for fulfilling concrete computing tasks, like for example processing properly the test Monte Carlo job. Based on the results of the experiment specific tests which are considered to be critical for a given activity or critical for the overall computing activities of a given VO, the site availability can be calculated and made available to site administrators. This is the purpose of the site availability application developed following the requirements of the CMS experiment.

The results of SAM tests are imported into the Experiment Dashboard database on a regular basis. The user interface provides the possibility to check the results of the latest tests, the latest site or service availability and the site or service availability as a function of time for a given time range. The first implementation of this application is currently in production. The next version should have an advanced interactive functionality, allowing to navigate from the site availability view to the page showing the availability of services belonging to a certain site and then to the results of the tests related to a given service.

4. Future plans

There are several directions regarding the future development of the Experiment Dashboard, but the main goal is to improve the reliability and completeness of the provided monitoring information. To achieve this goal we are adding new sources of the monitoring information. For example, for improving data reliability in the job monitoring application the Experiment Dashboard team is collaborating with Condor-G [16] developers in order to instrument the Condor-G submitter to report information related to job status changes to the Experiment Dashboard. At the same time, for jobs submitted via a gLite Resource Broker we explore the possibility to receive job status changes information from the gLite Logging and Bookkeeping system via a subscription mechanism.

The Experiment Dashboard is following the modifications of the work flow and data management organization of the LHC experiments. Some of the experiments are using pilot submission systems, where several user applications can run in a single Grid job. This is not currently supported by the Experiment Dashboard database schema. We are planning to modify the Experiment Dashboard schema and user interface to be able to cope with this submission mode.

We are working on improvements of the user interfaces, for example the enabling of secure access to certain types of information based on X509 authentication, where relevant.

Another important direction of the development is the improvement of effectiveness for troubleshooting. For instance, we analyze collected statistics on the job failures trying to define mechanisms which would allow to decouple application failures caused by the errors in the user code from the failures related to the problems of sites or Grid services.

Finally, we are working on the development of new applications, for instance, the production monitoring applications for ATLAS and CMS.

5. Conclusions

The Experiment Dashboard is currently in production for all four LHC experiments. The core part of the job monitoring functionality is also used by the Vlemed VO outside the LHC community. While most of the existing Grid monitoring tools are coupled to a given infrastructure or a given middleware, the Experiment Dashboard works across various Grid infrastructures. It uses multiple sources of monitoring data including other Grid monitoring systems or services specific to the experiments. This increases the reliability and completeness of the provided information.

The Experiment Dashboard framework defines the overall structure of the project, provides common ways for managing the components of the system and contains a lot of generic methods which can be used for the development of new applications. These functionalities of the framework facilitate a lot the new developments and everyday support of the system.

Close collaboration with the experiments allows to create the system focused on the exact needs of the users regarding monitoring, providing complete views of how the distributed infrastructure is used by the LHC experiments, and covering different areas of their activities.

6. Acknowledgements

The authors would like to express their gratitude to the developers of the monitoring, accounting and job submission systems, in particular MonALISA, Imperial College Real Time Monitoring, RGMA, GridIce, SAM, Apel, Gratia, and Condor-G for the fruitful collaboration, to our colleagues in the LHC experiments who contributed a lot to the progress of the Experiment Dashboard project, to the Oracle support team at CERN for excellent DB support and useful advices, and to EIS and FIO in CERN IT for their help and prompt response to our requests. Special thanks to Stefano Belforte - the initiator of the Experiment Dashboard project, Massimo Lamanna and Iosif Legrand without their support and valuable guidance the progress of the project would not be possible.

This work is funded by EGEE. EGEE is a project funded by the European Union under contract INFSO-RI-031688.

References

[1] LHC home page, http://lhc.web.cern.ch/lhc/

[2] The LCG Editorial Board, "LHC Computing Grid Technical Design Report", LCG-TDR-001, CERN-LHCC-2005-024, June 2005

[3] Enabling Grid for E-sciencE (EGEE), http://www.cern.ch/egee

[4] Open Science Grid (OSG) Web Page, http://www.opensciencegrid.org/

[5] Nordic Data Grid Facility (NDGF) Web Page, http://www.ndgf.org/

[6] R-GMA home page, http://www.r-gma.org/

[7] GridIce home page, http://gridice.forge.cnaf.infn.it//

[8] SAM home page, http://sam-docs.web.cern.ch/sam-docs/

[9] Imperial College Real Time Monitor home page, http://gridportal.hep.ph.ic.uk/rtm/

[10] Monitoring Agents Using a Large Integrated Services (MonALISA), http://monalisa.cern.ch/monalisa.html

[11] R. Rocha et al., "Monitoring the Atlas Distributed data Management System", Conference on Computing in High Energy and Nuclear Physics (CHEP07), September 2007, Victoria (Canada)

[12] VLEMED home page, http://www.science.uva.nl/~silvia/vlemed

[13] P. Saiz et al., "Grid reliability", Conference on Computing in High Energy and Nuclear Physics (CHEP07), September 2007, Victoria (Canada)

[14] S. Campana et al., "Experience with the gLite Workload Management System in ATLAS Monte Carlo Production on LCG, Conference on Computing in High Energy and Nuclear Physics (CHEP07), September 2007, Victoria (Canada)

[15] J. Hernandez Calama et al., "CMS Monte Carlo production in the WLCG Computing Grid", Conference on Computing in High Energy and Nuclear Physics (CHEP07), September 2007, Victoria (Canada)

[16] Condor-g home page, http://www.cs.wisc.edu/condor/condorg/