

# User-centric monitoring of the analysis and production activities within the ATLAS and CMS Virtual Organisations using the Experiment Dashboard system

---

## **Julia Andreeva**

*CERN, European Organization for Nuclear Research  
CH-1211, Genève 23, Switzerland  
E-mail: Julia.Andreeva@cern.ch*

## **Mattia Cinquilli**

*CERN, European Organization for Nuclear Research  
CH-1211, Genève 23, Switzerland  
E-mail: Mattia.Cinquilli@cern.ch*

## **Ivan Dzhunov**

*CERN, European Organization for Nuclear Research  
CH-1211, Genève 23, Switzerland  
E-mail: Ivan.Dzhunov@cern.ch*

## **Edward Karavakis<sup>1</sup>**

*CERN, European Organization for Nuclear Research  
CH-1211, Genève 23, Switzerland  
E-mail: Edward.Karavakis@cern.ch*

## **Michael Kenyon**

*CERN, European Organization for Nuclear Research  
CH-1211, Genève 23, Switzerland  
E-mail: mkenyon@cern.ch*

## **Lukasz Kokoszkiewicz**

*CERN, European Organization for Nuclear Research  
CH-1211, Genève 23, Switzerland  
E-mail: Lukasz.Kokoszkiewicz@cern.ch*

## **Pablo Saiz**

*CERN, European Organization for Nuclear Research  
CH-1211, Genève 23, Switzerland  
E-mail: Pablo.Saiz@cern.ch*

---

<sup>1</sup> Speaker

**Laura Sargsyan**

*ANSL (Yerevan Physics Institute) (AM)*

*E-mail: Laura.Sargsyan@cern.ch*

**David Tuckett**

*CERN, European Organization for Nuclear Research*

*CH-1211, Genève 23, Switzerland*

*E-mail: David.Tuckett@cern.ch*

Various fully distributed job submission methods and execution back-ends are used within both the ATLAS and CMS Virtual Organisations (VOs). More than 600,000 ATLAS and 400,000 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 (WMS), the Experiment Dashboard user-centric monitoring applications support different middleware implementations and job submission systems. The development of these applications was user driven with physicists invited to test the prototype versions in order to assemble further requirements and identify weaknesses with the applications.

*EGI Community Forum 2012 / EMI Second Technical Conference,  
Munich, Germany  
26-30 March, 2012*

## 1. Introduction

The Worldwide Large Hadron Collider Computing Grid (WLCG) [1] is a global collaboration between research centres and academic institutes that combines the power of more than 140 computing centres in 35 countries across the world. At any given time, there are more than 200,000 jobs running on behalf of the Large Hadron Collider (LHC) [2] Virtual Organisations (VOs). These jobs are running on various middleware flavours and they are being submitted with different submission methods and execution back-ends.

Job processing is the core part of the LHC VO computing activities as the physicists are eager to process, analyse and look at the data coming out of the LHC programme in search of new discoveries. The job processing activity is divided into two main categories:

- user analysis jobs
- data reconstruction and Monte-Carlo production jobs

While the latter is a well-organised activity performed by a group of experts that could easily identify errors and inefficiencies and correct them, the former is a chaotic activity performed by diverse members of the worldwide physics community who are not necessarily experienced in using the Grid.

A powerful and flexible monitoring application is required so that the scientists will be able to monitor the execution status, application and grid-level messages of their task that may be running at any site within the WLCG project. A powerful and flexible monitoring application would also bring an advantage to the user support team as only non-trivial issues will be escalated to them.

Most of the existing monitoring applications are coupled to VO-specific solutions. In CMS, there are at least five different submission tools and most of them provide their own monitoring system, for example, CRAB Monitoring is coupled to jobs submitted by the CRAB submission system [3] and WMAgent Monitoring is coupled to jobs submitted via the WMAgent workflow management tool [4]. Similarly, in ATLAS, there are two different workload submission systems that provide their own monitoring applications. Panda Monitoring is coupled to jobs submitted via the PanDA Workload Management System (WMS) [5] and Ganga Monitoring is coupled to jobs submitted from Ganga to either the gLite WMS or to a local batch system [6]. None of the previously mentioned systems are able to provide the global picture of the VO job processing activity.

The Experiment Dashboard [7] is a monitoring system developed for the LHC VOs in order to provide the view of the Grid infrastructure from the perspective of the VO. It enables a transparent view of the VO activities across different middleware implementations and combines Grid monitoring data with information that is specific to the VO. The Experiment Dashboard offers two Task Monitoring applications, one focused on the needs of analysis users and the other focused on the needs of production users. These two applications collect and expose a user-centric set of information to the user regarding submitted tasks. They provide a clear and precise view of the task status evolution and reason of failure as a function of time or site. The applications offer advanced failure diagnostics and detailed job information along with full resubmission history. Advanced graphical plots are also available which give a more usable

and attractive interface to the analysis and production user. These applications are based on common solutions and on a common database schema. Over the last three years, they were developed in close collaboration with the physicists who use the Grid infrastructure on a daily basis and as a result, they are tailored to their needs. This paper will

- describe the current job monitoring architecture of the Experiment Dashboard
- cover in detail the Experiment Dashboard system and the Task Monitoring applications that it offers for analysis and production LHC users
- and finally, provide an insight into future development plans.

## 2. Job Monitoring Architecture

The Experiment Dashboard monitoring system [8] has loose coupling to the information sources adding flexibility to the system. For job monitoring [9], it collects information from various information sources such as:

- the job submission systems of the VO
- jobs themselves running at the Worker Nodes (WNs)

For the monitoring of the job processing activity, it 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.

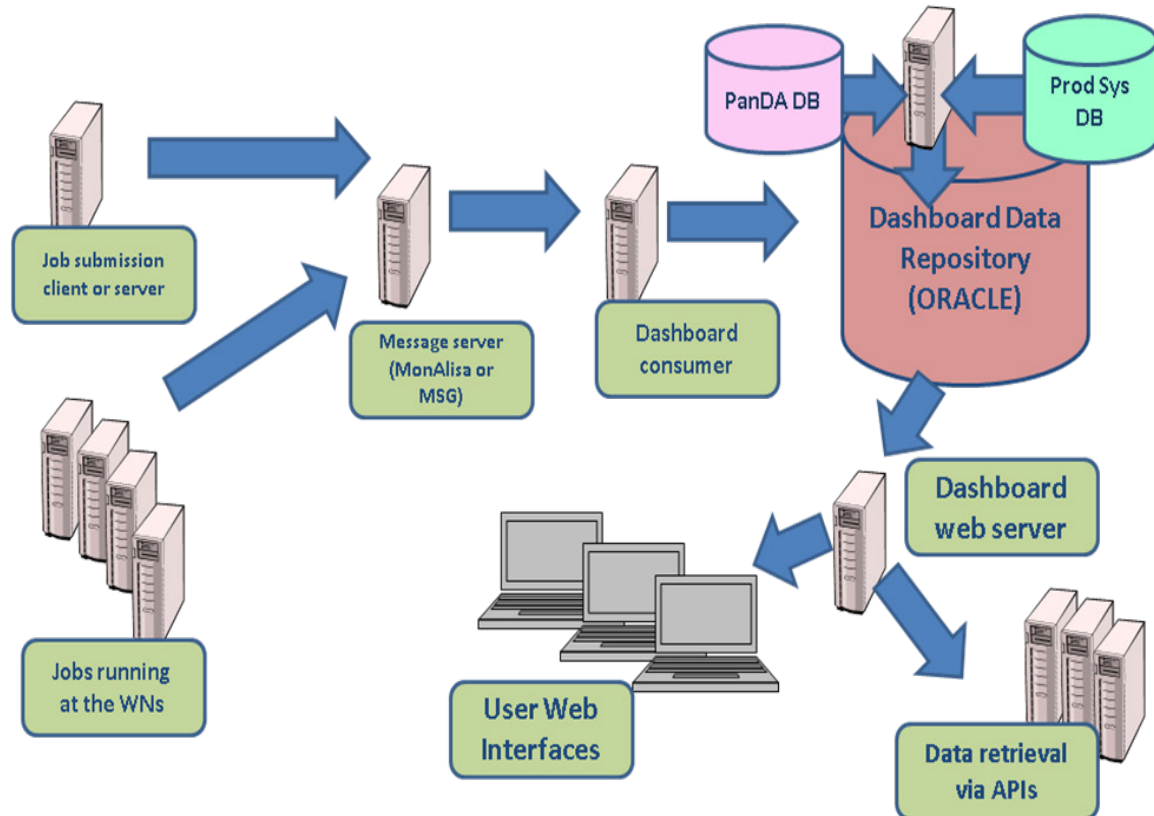


Figure 1 Experiment Dashboard job monitoring architecture

The job submission systems and the jobs running at the WNs are instrumented to report any job status updates to a messaging server that can be either the MonALISA monitoring system [10], used for all CMS jobs, or the Apache ActiveMQ Messaging System for the Grid (MSG) [11], used for ATLAS jobs submitted with Ganga [12]. 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 the 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 and then store and import it in the Dashboard Data Repository.

In order to ensure the reliability of the service, the Dashboard consumers are constantly monitored on multiple service levels:

- on the database level, with DB scheduled jobs checking the latest insertion time of the consumer and raising an alarm in case of a delay
- on the Operating System (OS) level, with Linux crontab scripts checking the status of the collector at frequent time intervals and restarting it if necessary
- on the consumer level, with code assertions checking error-conditions and raising an alarm in case of an error

The alarms are sent both as e-mails and as mobile text messages.

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 and CMS even though they are using different job submission systems and execution back-ends.

The User Web Interfaces are 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 user's 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 expose modern user-interfaces built with hBrowse [13], a common jQuery framework used for generic job monitoring applications.

The information stored in the Dashboard Data Repository is also consumed by other applications, either internal Dashboard applications or external third party ones, in machine-readable format through a set of well-defined APIs.

### **3. Experiment Dashboard Analysis Task Monitoring**

The Experiment Dashboard Analysis Task Monitoring application [14] exposes a user-friendly interface and offers a wide selection of plots. This application plays a very important role in the distributed analysis operations of both CMS and ATLAS as it empowers the users in such a way so that only serious issues are escalated to the user analysis 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, the application helps analysis support teams to solve user problems by offering advanced job and task debugging information.

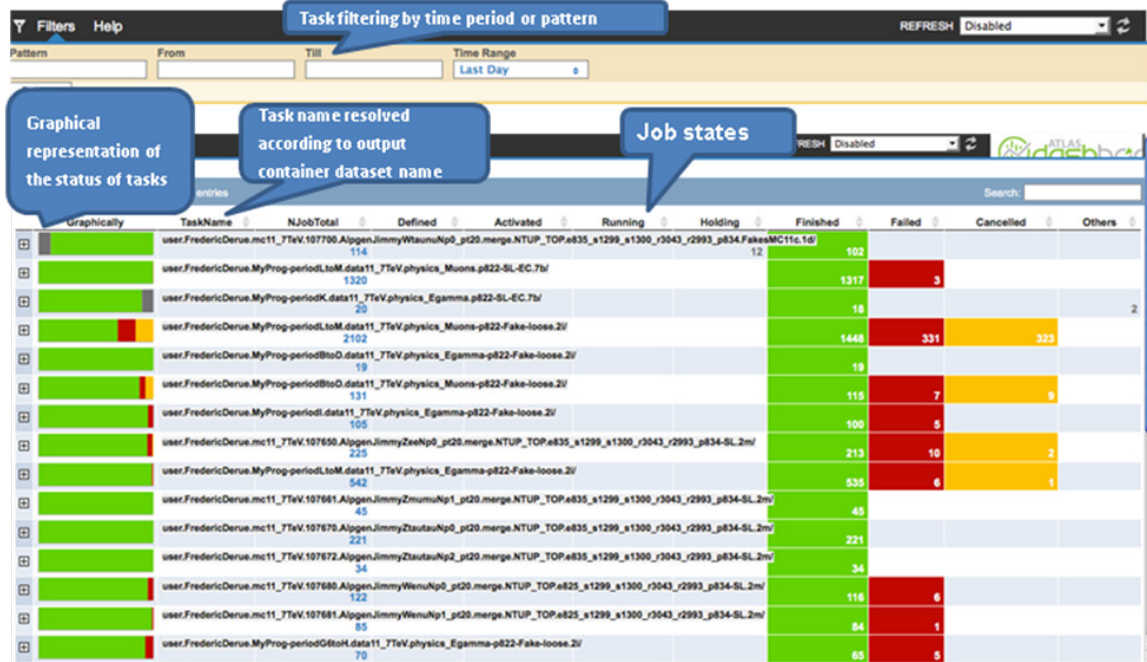


Figure 2 The user interface of the Analysis Task Monitoring

The application also offers “task meta information” (Figure 3) such as the creation time of the task, the last modification time, the input dataset used and the sites that the jobs of this particular task are running. Advanced interactive plots are also available and these can be exported as an image or PDF document.

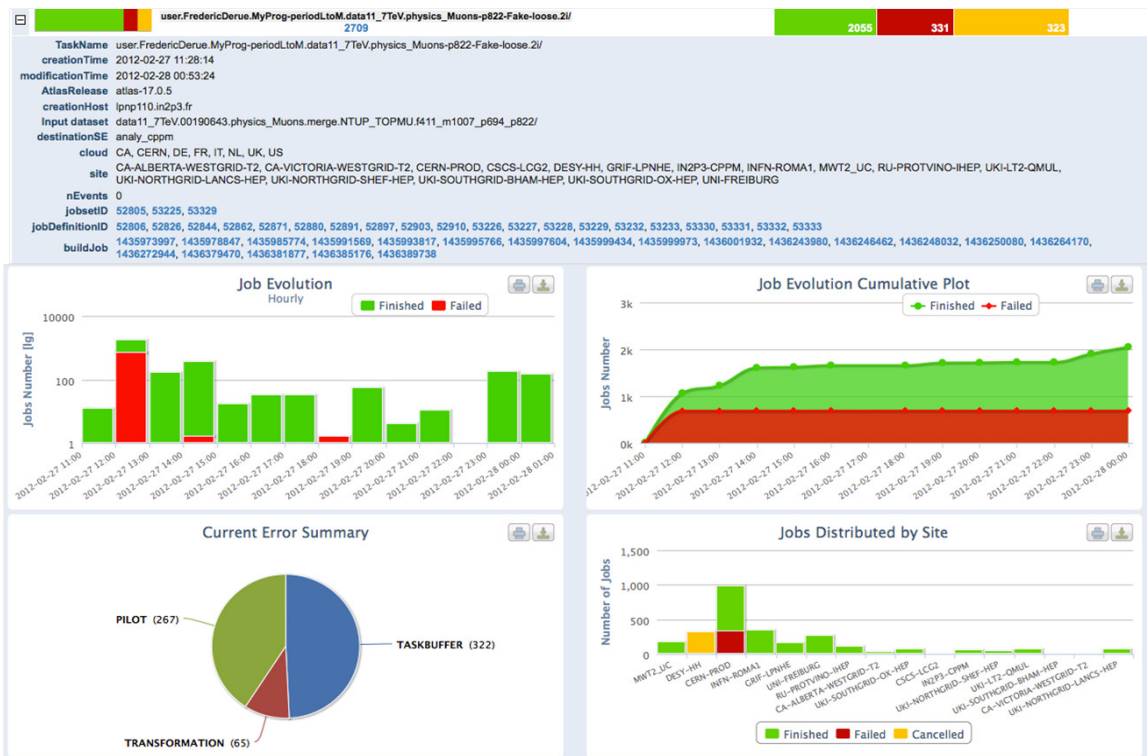


Figure 3 Task-meta information and advanced graphical plots

According to our Google Analytics web usage statistics, there has been an increase in the number of visitors using Android tablets and smart-phones on a daily basis. As a result, 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 (Figure 4). The popularity of the Android application has steadily increased. As of the first of January 2012, there were 2,866 visits from Android smart-phones and tablets out of 62,451 visits in total.



Figure 4 User Interface of the Android port of Analysis Task monitoring

#### 4. Experiment Dashboard Error Reporting Tool

Another very important application is the Error Reporting Tool that is heavily used by the CMS Analysis Operations Service. When a client-submitted job fails, the analysis user support team does not have access to the working directory of the user as it normally resides on the user's computer or on a remote Linux shell. The user can upload a snapshot of the working directory for investigation by the analysis operations team who use this information alongside detailed error diagnostics provided by the Analysis Task Monitoring application. The Error Reporting Tool was initially deployed as a VO independent Ganga service and subsequently adapted in collaboration with the CMS Analysis Operations team to meet the needs of CMS users. A snapshot of the user interface of the application can be seen in Figure 5. This approach significantly optimises the workflow of the analysis support team and thus, it has become essential to their everyday operations.

**CMS error reports**

Username:  Version:  From:  Till:

Reports

Show 25 entries

Report number	Time uploaded	Username	Version	Monitoring link	Download
1302	2012-03-18 18:52:25	ventura	CRAB_2.8.0	Dashboard Task Mon CrabServer Logana CrabServer Status	Download
1301	2012-03-18 18:52:16	ventura	CRAB_2.8.0	Dashboard Task Mon CrabServer Logana CrabServer Status	Download
1300	2012-03-18 18:53:07	ventura	CRAB_2.8.0	Dashboard Task Mon CrabServer Logana CrabServer Status	Download
1299	2012-03-18 11:09:07	nkumar	CRAB_2.8.0	Dashboard Task Mon	Download
1298	2012-03-18 11:08:09	nkumar	CRAB_2.8.0	Dashboard Task Mon	Download
1297	2012-03-18 11:07:44	nkumar	CRAB_2.8.0	Dashboard Task Mon	Download
1296	2012-03-18 11:04:54	nkumar	CRAB_2.8.0	Dashboard Task Mon	Download
1295	2012-03-17 10:00:41	sudha	CRAB_2.7.9	Dashboard Task Mon	Download
1294	2012-03-16 17:19:37	jojob	CRAB_2.8.0	Dashboard Task Mon CrabServer Logana CrabServer Status	Download
1293	2012-03-16 17:05:19	jojob	CRAB_2.8.0	Dashboard Task Mon CrabServer Logana CrabServer Status	Download
1292	2012-03-16 08:53:32	ventura	CRAB_2.8.0	Dashboard Task Mon	Download
1291	2012-03-16 08:52:27	ventura	CRAB_2.8.0	Dashboard Task Mon	Download
1290	2012-03-16 08:52:04	ventura	CRAB_2.8.0	Dashboard Task Mon	Download

Figure 5 CMS Error Reporting Tool

## 5. Experiment Dashboard Production Task Monitoring

The production teams of the LHC VOs are responsible for data reconstruction and Monte-Carlo production jobs. They have to follow the processing of hundreds of thousands of jobs which normally run across numerous computing sites. Therefore, the monitoring and book-keeping of production jobs is a challenging task. The Production Task Monitoring application allows production managers to follow the progress of their tasks by offering a wide selection of statistics and graphical plots. By using this tool, production managers can follow the processing of the production workflows and can easily detect any inefficiencies and/or delays in the execution of their production tasks. The application takes into account feedback collected from the ATLAS production managers.

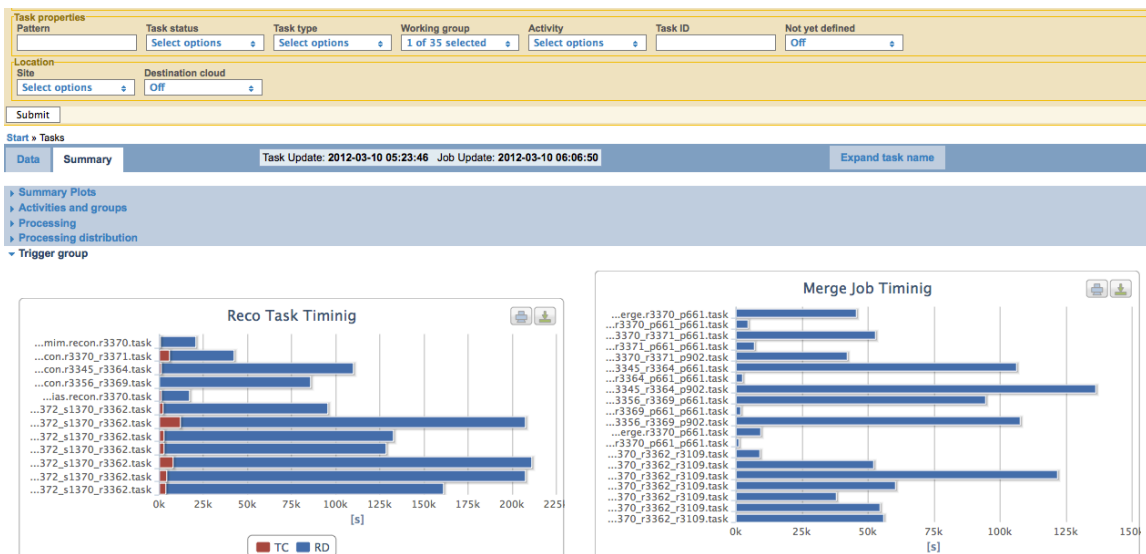


Figure 6 The user interface of the Production Task Monitoring



## 6. Future Development

The Experiment Dashboard job monitoring applications will be extended according to the requests of the LHC VOs. The Analysis Task Monitoring application will become more interactive as it will support the resubmission and cancellation of a given task or a given job and/or a subset of jobs directly from the web user interface. A prototype version of the Analysis Task Monitoring with this feature enabled has been recently released and it is being extensively tested by selected power users of the ATLAS VO community to validate the functionality and by security experts at CERN to check for any vulnerabilities. Currently the users are authenticated with their Grid Certificate over a secure HTTP connection. The Production Task Monitoring will be extended according to the requests that are being collected from ATLAS production managers.

## 7. Conclusions

The Experiment Dashboard Task Monitoring applications for analysis and production users have become very popular within the ATLAS and CMS user communities and play an important role in the analysis and production operations of the LHC. They also play an important role in the support infrastructure [15, 16, 17] as they ensure that only serious issues are escalated to the support teams. During the first year of LHC data taking, the Experiment Dashboard 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 Experiment Dashboard Task Monitoring applications for analysis and production users offer advanced, intuitive and user-centric web interfaces by providing a clear and precise view of task status evolution. They offer advanced failure diagnostics, detailed job information along with full resubmission history. A wide selection of advanced graphical plots are also available which give a more attractive interface to the analysis and production user. The job monitoring architecture and Experiment Dashboard Task Monitoring applications are shared between ATLAS and CMS even though they use different job submission systems, execution backends, and information systems. This is made possible thanks to the approach of using common solutions within the Experiment Dashboard and a shared database schema between the two VOs.

There has been significant progress in the development of applications for monitoring of user analysis and production activities within the Experiment Dashboard framework from 2009 onwards. This work is very important, since it contributes to the overall success of the LHC offline computing effort. Regular consistency checks against the job submission systems of ATLAS and CMS, as the primary information sources, ensure the reliability of monitoring data.

These applications are being developed in very close collaboration with the physicists who use the Grid infrastructure to submit analysis and production jobs and as a result, they respond well to the needs of the LHC user community.

## References

- [1] J. Shiers, *The Worldwide LHC Computing Grid (worldwide LCG)*, 2006, Proc. of the Conference on Computational Physics (CCP06) Computer Physics communications **177**, pp 219–223  
[doi:10.1016/j.cpc.2007.02.021](https://doi.org/10.1016/j.cpc.2007.02.021)
- [2] L. Evans and P. Bryant, *LHC Machine*, 2008, JINST **3** S08001 [doi:10.1088/1748-0221/3/08/S08001](https://doi.org/10.1088/1748-0221/3/08/S08001)
- [3] D. Spiga, M. Cinquilli et al, *CRAB: an application for distributed scientific analysis in grid projects*, 2008, International Conference on Grid Computing & Application – Worldcomp, GCA2008:187-193
- [4] F. van Lingen et al, *Job life cycle management libraries for CMS workflow management projects*, 2010, J. Phys.: Conf. Ser. **219** 042024 [doi:10.1088/1742-6596/219/4/042024](https://doi.org/10.1088/1742-6596/219/4/042024)
- [5] T. Maeno et al, *Overview of ATLAS PanDA Workload Management*, 2011 J. Phys.: Conf. Ser. **331** 072024 [doi:10.1088/1742-6596/331/7/072024](https://doi.org/10.1088/1742-6596/331/7/072024)
- [6] J. Elmsheuser et al, *Distributed analysis in ATLAS using GANGA*, 2010, J. Phys.: Conf. Ser. **219** 072002 [doi:10.1088/1742-6596/219/7/072002](https://doi.org/10.1088/1742-6596/219/7/072002)
- [7] 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](https://doi.org/10.1088/1742-6596/331/7/072001)
- [8] J. Andreeva et al, *Experiment Dashboard for monitoring computing activities of the LHC virtual organizations*, 2010, J. Grid Comput. **8** 323-339 [doi:10.1007/s10723-010-9148-x](https://doi.org/10.1007/s10723-010-9148-x)
- [9] J. Andreeva et al, *Job monitoring on the WLCG scope: Current status and new strategy*, 2010, J. Phys.: Conf. Ser. **219** 062002 [doi:10.1088/1742-6596/219/6/062002](https://doi.org/10.1088/1742-6596/219/6/062002)
- [10] 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](https://doi.org/10.1016/j.cpc.2009.08.003)
- [11] 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](https://doi.org/10.1088/1742-6596/219/7/072017)
- [12] 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 (CHEP'12)
- [13] 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](https://doi.org/10.1016/j.cpc.2012.08.062)
- [14] 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](https://doi.org/10.1088/1742-6596/219/7/072038)
- [15] A. Fanfani et al, *Distributed analysis in CMS*, J. Grid Comput. **8** 159-179 [doi:10.1007/s10723-010-9152-1](https://doi.org/10.1007/s10723-010-9152-1)
- [16] J. Andreeva et al, *CMS analysis operations*, 2010, J. Phys.: Conf. Ser. **219** 072007  
[doi:10.1088/1742-6596/219/7/072007](https://doi.org/10.1088/1742-6596/219/7/072007)
- [17] J. Schovancova et al, *ATLAS Distributed Computing Monitoring tools after full 2 years of LHC data taking*, 2012, to appear in the proceedings of Computing in High Energy and Nuclear Physics (CHEP'12)