CMS Dashboard for Monitoring of the User Analysis Activities

Edward Karavakis[a,b], Julia Andreeva[b], Gerhild Maier[b] and Akram Khan[a]

[a]*School of Engineering and Design, Brunel University, Uxbridge, UB8 3PH, UK*
[b]*CERN, European Organisation for Nuclear Research, CH-1211 Geneva 23, Switzerland*

Abstract. The CMS Virtual Organisation (VO) uses various fully distributed job submission methods and execution backends. The CMS jobs are processed on several middleware platforms such as the gLite, the ARC and the OSG. Up to 200,000 CMS jobs are submitted daily to the Worldwide LHC Computing Grid (WLCG) infrastructure and this number is steadily growing. These mentioned factors increase the complexity of the monitoring of the user analysis activities within the CMS VO. Reliable monitoring is an aspect of particular importance; it is a vital factor for the overall improvement of the quality of the CMS VO infrastructure.

Keywords: Distributed Analysis in CMS, CMS Dashboard, CMS Monitoring.

INTRODUCTION

Distributed analysis on the WLCG infrastructure is currently one of the main challenges of the LHC computing. Transparent access to the LHC data has to be provided for more than 5000 scientists all over the world. Users who run analysis jobs on th Grid do not necessarily have expertise in Grid computing. Currently, 100-150 distinct CMS users submit their analysis jobs to the WLCG daily. The significant streamlining of operations and the simplification of end-users' interaction with the Grid are required for effective organisation of the LHC user analysis. Simple, user-friendly, reliable monitoring of the analysis jobs is one of the key components of the operations of the distributed analysis. Such monitoring is required not only for the physicists running analysis jobs but also for the analysis support teams.

Most of the CMS analysis users interact with the Grid via the CMS Remote Analysis Builder (CRAB). User analysis jobs can be submitted either directly to the WLCG infrastructure or via the CRAB analysis server, which operates on behalf of the user. In the first case, the support team does not have access to the log files of the user's job or to the CRAB working directory, which keeps track of the task generation. To understand the reason of the problem of a particular user's task, the support team needs a monitoring system capable of providing complete information about the task processing. To serve the needs of the analysis community and of the analysis support team, the CMS Dashboard Task Monitoring[1] and CMS Data Mining[2] applications have been developed on top of the CMS job monitoring repository.

This paper describes the framework of the Experiment Dashboard monitoring system, covers the CMS Dashboard applications which are widely used by the CMS physicists community, and provides an insight into future development plans.

FRAMEWORK

The common structure of the Experiment Dashboard service consists of information collectors, data repositories, normally implemented in ORACLE databases, and user interfaces. The Experiment Dashboard uses multiple sources of information[3]. The following information sources are being used for the job monitoring application:

- The Imperial College Real Time Monitor (ICRTM)[4]
- gLite Grid services, such as the Logging and Bookkeeping service (LB)[5] or CEMon[6]
- Distributed Production Agents for CMS
- Experiment central databases such as the PANDA database for ATLAS)

- Experiment client tools for job submission such as Ganga[7] for ATLAS and CRAB[8] for CMS
- Jobs instrumented to report directly to the Experiment Dashboard

This list is not exhaustive. Information can be transported from the data sources via various protocols. In most cases, the Experiment Dashboard uses asynchronous communication between the source and the data repository. For several years, in absence of a messaging system as a standard component of the gLite middleware stack, the MonALISA[9] monitoring system was successfully used as a messaging system for the Experiment Dashboard job monitoring applications. Currently, the Experiment Dashboard is being instrumented to use the Messaging System for the Grid (MSG)[10] for the communication with the information sources.

A common framework providing components for the most usual tasks was established to fulfill the needs of the Dashboard applications being developed for all experiments. The schema of the Experiment Dashboard framework is presented in Figure 1.

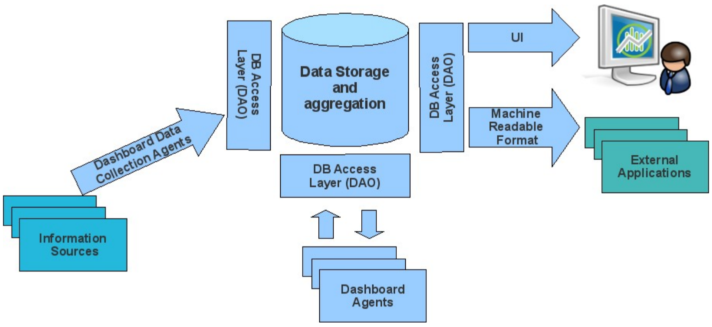

FIGURE 1. Experiment Dashboard Framework Schema.

CMS TASK MONITORING

The CMS Dashboard Task Monitoring[1] application exposes a user-centric set of information to the user regarding submitted tasks. It provides a clear and precise view of the status of the jobs submitted to the Grid with very low latency. The application provides monitoring functionality regardless of the job submission method or the middleware flavour and it works transparently across various Grid infrastructures[11]. CMS jobs are instrumented to report their status to the Dashboard in real-time and the job submission tools of the CMS experiment and the job wrappers generated by these tools are also instrumented to report task meta-information such as the creation time of the task, the input data collection and the number of events to be processed by a single job. 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. A selection of snapshots of the application can be seen in Figure 2.

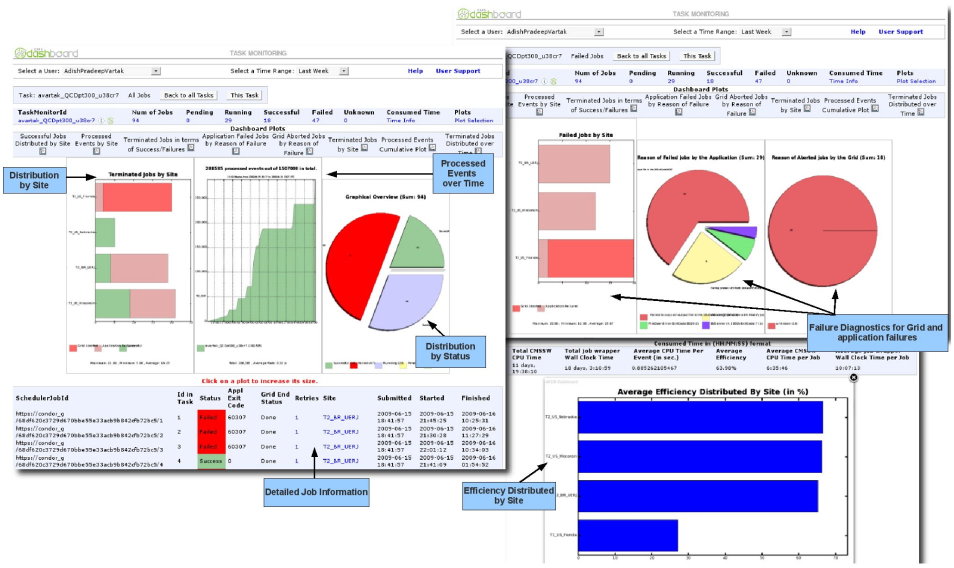

FIGURE 2. Selection of snapshots of the CMS Task Monitoring application.

The application offers a wide variety of graphical plots that will visually assist the user to understand the status of the task by providing a more attractive and usable user-interface. These plots show the distribution by site of successful, failed, running and pending jobs as well as for the processed events and they can help identify any problematic site and blacklist it from further resubmissions. They also demonstrate the terminated jobs in terms of success or failure and over the time range that the task has been running. 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 such as the distribution of CPU and wall clock time spent on successful and on failed jobs and the average efficiency distributed by site. These plots 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 development was user-driven with physicists invited to test the prototype in order to assemble further requirements and identify any weaknesses with the application. Close collaboration with several CMS users resulted in the tool being focused on their exact monitoring needs containing no unnecessary information. The monitoring tool has become very popular among the CMS users. According to our web statistics[12], more than one hundred distinct analysis users are using it for their everyday work.

CMS DATA MINING

Job failures on the Grid infrastructure can be caused by a variety of reasons, among them errors in the user code, corrupted input data, faulty experiment-specific software distribution at the site, failure of the Grid services, misconfiguration of the worker nodes at the site, expiration of the user proxy and many others. Understanding the actual failure reason is a difficult task. Association rule mining is applied to the job monitoring data to automatically retrieve knowledge about the behavior of Grid components by taking dependencies between job characteristics into account[2]. Therewith, problematic components are located automatically and this information, expressed by association rules, is visualised in a web interface. In order to present the mined information, a web interface was developed, which assists the CMS analysis support group at detecting problems with the CMS analysis jobs running on the WLCG infrastructure. Figure 3 depicts the QAOES web interface with one association rule indicating that a specific site is problematic.

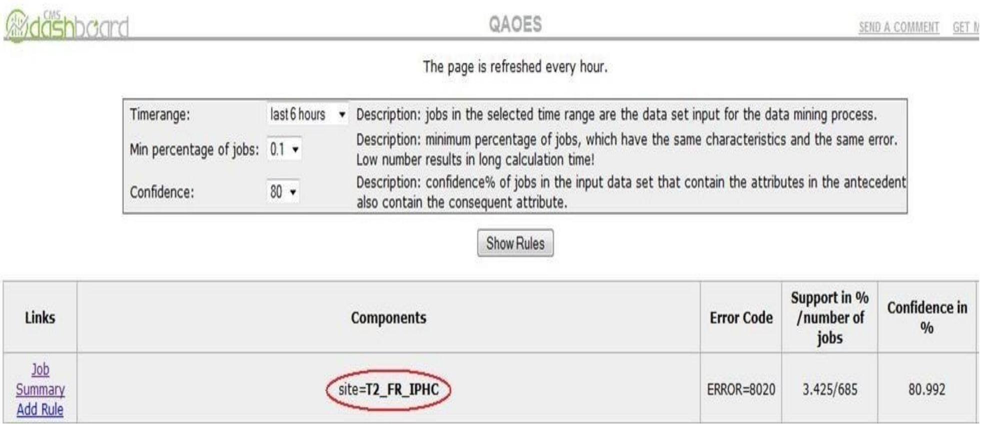

FIGURE 3. The Quick Analysis Of Error Sources (QAOES) interface.

FUTURE WORK

The future evolution of the user analysis applications is driven by the requirements of the CMS community which is preparing for the LHC data taking at the end of 2009. One of the improvements foreseen for the CMS Task Monitoring application is to improve the failure diagnostics for both Grid and application failures. The ideal situation would be to reach to a point where the user shouldn't have to open the log file to search for what went wrong. The user should get everything from the monitoring tool. A development effort is ongoing to improve the failure diagnostics reported to the Dashboard from the job wrapper. Due to the increased success of the CMS Task Monitoring application within the CMS community, there are plans to adapt it to the ATLAS VO as well.

Future development for the CMS Data Mining application foresees the implementation of a system to collect expert interpretations about the detected problematic Grid components in order to formulate generalised rules about the actual fault and a possible solution. These rules will form a knowledge base, which is consulted in real time whenever a new association rule is generated.

CONCLUSION

The Experiment Dashboard system was proven to be an essential component for the LHC computing operations. The variety of its applications covers the full range of the LHC computing activities. The system is being developed in a very close collaboration with the physicists who use the Grid infrastructure to analyse physics data. As a result, the Experiment Dashboard responds well to the needs of the LHC experiments.

There was a big progress in the development of applications for monitoring the user analysis activities during the year of 2009. This work is very important, since it contributes to the overall success of the LHC offline computing. The behaviour of analysis jobs is particularly difficult to predict, as it is a chaotic activity carried out by users who do not have to be necessarily experienced in using the Grid and locating problems themselves.

While other monitoring tools concentrate on a specific middleware, the CMS Dashboard applications provide monitoring functionalities regardless of the job submission method or the middleware platform.

In addition to the web applications providing the collected monitoring information, the Experiment Dashboard also offers a probabilistic approach to reveal hidden information in the monitoring data about faulty Grid components, by taking dependencies between job characteristics into account.

ACKNOWLEDGMENTS

The authors would like to thank all the participating users of the CMS Community for their feedback and support. This work was funded by EGEE. EGEE is a project funded by the European Union under contract INFSO-RI-031688.

REFERENCES

1. E. Karavakis, “CMS Dashboard Task Monitoring: A User-Centric Monitoring View”, CHEP'09: Proceedings of the 2009 International Conference on Computing in High Energy and Nuclear Physics.
2. G. Maier et al, “Association Rule Mining on Grid Monitoring Data to Detect Error Sources”, CHEP'09: Proceedings of the 2009 International Conference on Computing in High Energy and Nuclear Physics.
3. J. Andreeva et al, “Job Monitoring on the WLCG Scope. Current Status and New Strategy”, CHEP'09: Proceedings of the 2009 International Conference on Computing in High Energy and Nuclear Physics.
4. Imperial College Real Time Monitoring (ICRTM), http://gridportal.hep.ph.ic.ac.uk/rtm/
5. LB Homepage, http://egee.cesnet.cz/cz/JRA1/LB/
6. C. Aiftimiei et al, “Using CREAM and CEMON for Job Submission and Management in the gLite Middleware”, CHEP'09: Proceedings of the 2009 International Conference on Computing in High Energy and Nuclear Physics.
7. J. Moscicki et al, “Ganga: A Tool for Computational-Task Management and Easy Access to Grid Resources”, Computer Physics Communication, arXiv:0902.2685v1.
8. D. Spiga et al, “The CMS Remote Analysis Builder (CRAB)”. Lecture Notes in Computer Science, 2007.
9. Monitoring Agents Using a Large Integrated Services Architecture (MonALISA), http://monalisa.cern.ch/monalisa.html
10. J. Casey et al, “Monitoring the Efficiency of User Jobs”, CHEP'09: Proceedings of the 2009 International Conference on Computing in High Energy and Nuclear Physics.
11. J. Andreeva et al, “Experiment Dashboard: The Monitoring System for the LHC Experiments”, GMW'07: Proceedings of the 2007 Workshop on Grid Monitoring.
12. Dashboard Application Usage Statistics, http://lxarda18.cern.ch/usage.html