PROCEEDINGS OF SCIENCE

Automating CMS Calibrations using the WMAgent framework

Hassen Riahi[*†]
INFN and Università di Perugia
E-mail: hassen.riahi@pg.infn.it

Daniele Spiga
CERN
E-mail: daniele.spiga@cern.ch

Simon Metson
Bristol University
E-mail: simon.metson@cern.ch

Dave Evans
FNAL
E-mail: evansde@fnal.gov

Steve Foulkes
FNAL
E-mail: sfoulkes@fnal.gov

James Jackson
Bristol University and Rutherford Appleton Laboratory
E-mail: james.jackson@cern.ch

Giuseppe Codispoti
INFN and Università di Bologna
E-mail: giuseppe.codispoti@bo.infn.it

Mattia Cinquilli
INFN and Università di Perugia
E-mail: mattia.cinquilli@pg.infn.it

Fabio Farina
INFN Milano Bicocca
E-mail: fabio.farina@cern.ch

PoS(ACAT2010)069

PROCEEDINGS OF SCIENCE

Particle beams are now circulating in the world's most powerful particle accelerator LHC at CERN and the experiments are ready to record data from beam. Data from first collisions will be crucial for sub-detector commissioning, making alignment and calibration high priority activities. Executing the alignment and calibration workflows represent a complex and time consuming task, with intricate data dependencies and stringent latency requirements. For this reason CMS Workload Management has defined an architecture and strategy to automate the workflows to minimise the required human effort.

In this paper, we discuss the WMCore and WMAgent components used in the current CRABSERVER framework showing the prototype tuned over the first use case, namely calibration of the Beam Spot. We also present the longer term strategies for moving alignment and calibration workflows into the standard Tier-0 processing and automating the analysis workflow.

13th International Workshop on Advanced Computing and Analysis Techniques in Physics Research
February 22-27, 2010
Jaipur, India

*Speaker.
†Presented on behalf of the CMS collaboraton.

PoS(ACAT2010)069

1. Introduction

The central goal of the Compact Muon Solenoid (CMS) experiment[1] is to explore physics at the TeV energy scale, exploiting the collisions delivered by the Large Hadron Collider (LHC) at CERN[2]. The CMS trigger and data acquisition (DAQ) system is designed to collect and analyze the detector information at the LHC bunch-crossing frequency of 40 MHz. The rate of events to be recorded for offline processing and analysis is about a few hundred Hz. The trigger system reduces the event frequency from 1 GHz to $\sim$ 300 Hz by selecting the events of potential interest.

The first-level trigger (L1) is designed to reduce the incoming data rate to a maximum of 100 kHz, by processing fast trigger information coming from the calorimeters and the muon chambers, and selecting events with interesting signatures. This is further reduced by a factor of 1000 using a high-level trigger (HLT), a software filtering system running on a large processor farm, for selecting events to be accepted for storage. In the end, storage managers (SMs), stream event data to disk and transfer complete data files to the main CERN computing centre (Tier-0) for further processing offline. Routing of individual event data to files in the SM is driven by the definition of output streams, which group events selected by specific HLT paths. All streams defined by the online system and the HLT are written in a binary data format, referred to as streamer files. Streamer files are copied from the online systems at the detector site to the Tier-0, where they are converted into the ROOT-based event data format and repacked into primary datasets according to their physics content.

While some calibrations are performed online at the CMS detector site, the offline workflows for alignment and calibration are performed at the Tier-0 site and at the CERN Analysis Facility (CAF). They are used to compute and improve the alignment and calibration constants. Event information relevant for alignment and calibration is streamed from the CMS detector site via the standard physics event stream and a special calibration stream, as well as streams with special event content. The reconstructed data are then skimmed to create a series of datasets that contain only the minimal amount of information required by the alignment and calibration workflows (AlCaReco). These datasets are transferred to the CAF where they are processed by the alignment and calibration algorithms.

The alignment and calibration workflows, performed at the CAF, use the AlCaReco datasets to generate alignment and calibration constants that are validated and uploaded to the Conditions Database. Re-reconstruction at the Tier-1 sites, using the new constants, also generates new AlCaReco datasets that are used in turn as input to the next series of improvements on alignment and calibration constants.

Executing the offline workflows for alignment and calibration represents a complex and time consuming task, with intricate data dependencies and stringent latency requirements. In addition, there is a need to reduce the delays executing the typical analysis workflows in order to be able to run analysis jobs as soon as data are available at the various computing sites. These are the main motivations which led CMS to develop a fully automated system for managing analysis and calibration workflows. This system is a tool designed to cache and prioritise user requests, feeding information to components that take care of the distributed data bookkeeping, and then triggering the submission of jobs once user conditions are satisfied.

In this paper, we discuss the design details of the automation system, introducing briefly the

tools and terminologies. We also present the motivations and the details of the integration strategy for the tools. We show the automation system tuned over the first use case, the calibration of the Beam Spot measurement[3]. Finally, we report the short term CMS strategy to automate data analysis workflows, as well as the longer term strategy for moving alignment and calibration workflows into the Tier-0 standard processing.

2. Requirements for automated processing of alignement and calibration workflows

The goal of this work is to provide an automatic way of managing CMS data analysis workflows. The basic concept of the automation is that workflows should be executed in a data driven manner. It foresees to take actions automatically when a sufficient amount of suitable new data are ready. The actions typically involve submitting jobs as soon as sufficient new data are accumulated, which can be defined as a number of new files or new luminosity sections in a run, the end of a run, or as a given number of events in a file etc.

As already mentioned, the alignment and calibration workflows can be complex since the outputs of one execution step serves as input of the next and thus can be represented symbolically via a a Direct Acyclic Graph (DAG) model (shown in Figure 1).

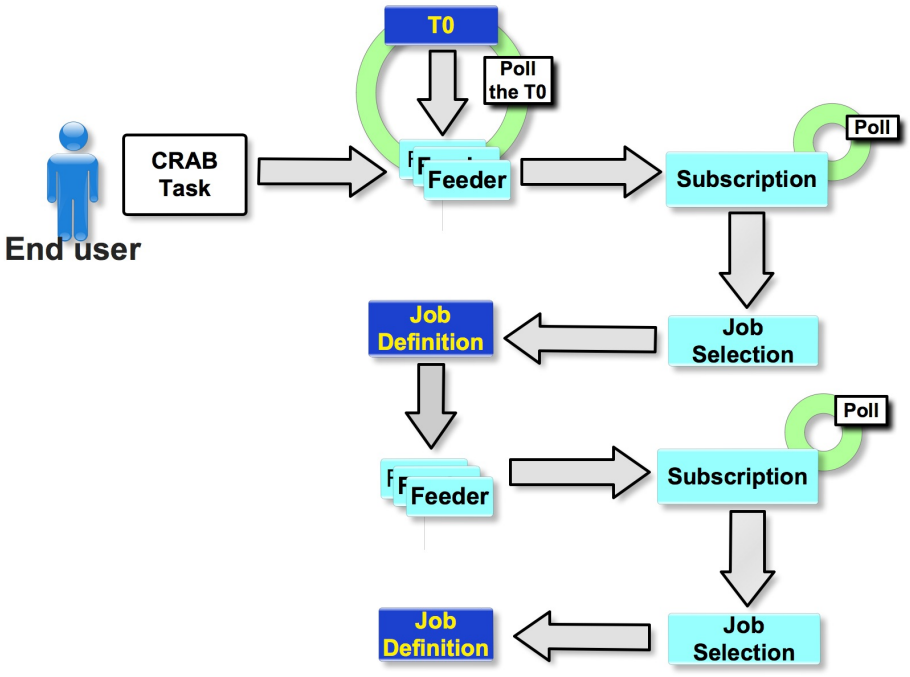

Figure 1: Complex automated workflows

The foremost objective is to automate alignment and calibration workflows for the following reasons:

- during the initial data-taking, the workflows are not mature enough to be handed over to a central data operations team;
- the alignment and calibration team need to be able to monitor the execution of their workflows very closely and apply correctness to the methods themselves and their configurations;
- and with less priority, to reduce the delays supported by typical analysis workflows.

The general aim is to automate more the work done on behalf of users reducing the efforts that they invest to manage their analysis workflows and decreasing, of course, the delays in their execution.

3. Implementation strategy

The architecture of the new CMS Workload Management system is based on the new WMCore/WMAgent frameworks [4] and makes use of existing CMS tools, in particular the CMS Remote Analysis Builder (CRAB)[5].

WMCore/WMAgent is the common workload management platform, including core libraries, for managing jobs in CMS. It is designed to support three main application areas: data processing and reconstruction, large scale production of Monte Carlo simulated events and data analysis. CRAB is built on a client-server model that interfaces the user to the distributed computing environment [6]. In the future, it is intended that CRAB itself will be implemented on top of these new common workload management components. Finally, one of the key ingredients in the integration of the above mentioned toolkits is BossLite[7], a Python library developed to interface the Workload Management tools to the grid middleware and local batch systems.

In order to build the infrastructure needed to support automated workflows, an interface has been developed in the current CRABSERVER framework to interact with WMAgent using WMBS (Workload Management Bookeeping system).

The key feature of the design strategy is to exploit the WMBS system in CRABSERVER. WMBS is a library provided as part of WMCore framework, which is by design fully usable as part of the Tier-0 architecture. This approach is fully exploited to realise the migration of alignment and calibration workflows into the Tier-0 system. The use of CRAB allows us to benefit from the simplicity of its existing Grid interface and also to achieve the first step required to build the next generation of CRAB framework (based on WMCore/WMAgent) . Figure 2 shows a simple automated workflow using CRABSERVER and WMBS.

The most important objects of the WMBS system used to automate workflows are WMBS instance, WMBS Subscription, and WMBS Feeder:

- WMBS instance is a workflow dependency manager that can be used to drive subsequent processing steps on the output of jobs[8] as they run, such as merging.
- An instance of WMBS Feeder watches a data source and injects new data into the Subscription. A Feeder can be Dataset Bookkeeping System (DBS)[9], T0AST, Phedex[10], UserPlugin, etc.

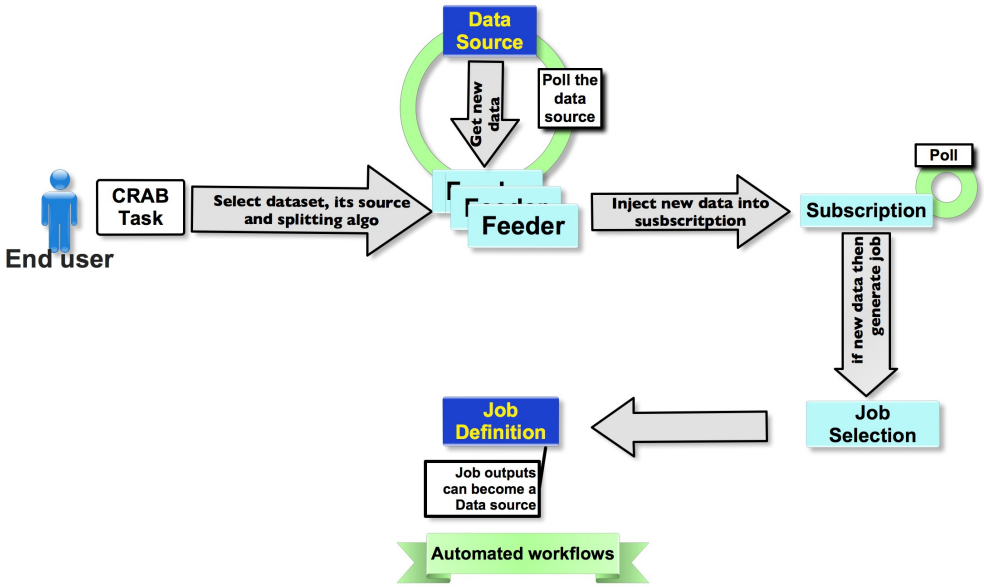

Figure 2: Simple automated workflows using CRABSERVER and WMBS

- WMBS Subscription is an association of a WMBS Workflow, where a splitting algorithm is defined, and a WMBS Fileset. WMBS Subscription watches new slices of data that accumulates over the time. Active subscriptions are checked for new files and if conditions are met, new jobs are created according to the algorithm defined in the WMBS Workflow.

4. Integration of CRABSERVER and WMBS

Both CRABSERVER and WMAgent implement a modular architecture, based on independent agents communicating through a persistent message system, based on a database. Within the automation system, modules from CRAB and WMAgent are involved to provide a flexible WMBS Feeder based infrastructure. In this way, requirements of different kinds of workflows can be fulfilled by customizing dedicated Feeders.

CRAB Client provides a simple user interface for the submission of workflows. From the workflow point of view, it transmits task specifications encoded in XML[11] to CRABSERVER for processing.

Once the task specification reaches CRABSERVER, the TaskRegister component starts the processing. It adopts a thread-based architecture to split the work in order to satisfy efficiently the large number of tasks sent by CMS users. It creates BossLite objects and therefore contacts the FeederManager and WorkflowManager through the WMBS MsgService object. It sends an "AddDatasetWatch" message to the FeederManager to instantiate WMBS objects, namely WMBS Feeder and WMBS Fileset, and an "AddWorkflowToManage" message to the WorkflowManager to create a WMBS Workflow object.

The FeederManager associates WMBS Fileset and WMBS Feeder objects. A new thread is handled in this component to create an empty Fileset once the "AddDatasetWatch" message is received from the TaskRegister. Furthermore, it polls regularly, in a main thread, the data source

across the WMBS Feeders to complete file information of managed Filesets. The Feeder closes the Fileset when all its files are there. The format of messages coded in Python and the details of the "AddDatasetWatch" message are shown in Figure 3.

{'name':'AddDatasetWatch', 'payload':{'FeederType':feeder, 'dataset':dataset,
'FileType':processing, 'StartRun':startRun}}

Figure 3: Messages format

The WorkflowManager associates WMBS Workflow and WMBS Fileset objects. In the same way as the FeederManager, this component polls the WMBS database regularly to check unsubscribed Fileset to be matched by the managed Workflows and vice versa. The WMBS Workflow name is a regular expression specified by the user in the CRAB configuration to specify the data that he is interested in. The WorkflowManager creates a Subscription if it matches a Fileset with a Workflow. A user can also remove Workflows from CRABSERVER whenever they wish.

At this point in the workflow, all information needed to create new BossLite jobs is available. CrabJobCreator is the component developed in CRABSERVER to support the integration and the automation system. It polls existing Subscription objects to check new data. Subscriptions acquiring new files are picked up to take the following actions:

1. Applies the WMBS splitting algorithm to the Subscription to create WMBS Jobs .
2. Completes the BossLite Task filling BossLite Job and runinngJob tables.
3. Contacts CrabServerWorker for jobs submission.

These modules are shown schematically in Figure 4.

5. Description of the first supported workflow

The first workflow that is being supported by the automation system is the beam spot determination. The beam spot is the luminous region produced by the collisions of the LHC proton beams. Knowing the position of this is important for accurate physics measurements, and it needs to be measured precisely in an automated fashion for a correct offline data reconstruction [12]. The new section in CRAB configuration file, named [WMBS], is used to specify all configurable parameters needed to drive the automatic process (see Figure 5).

For the beam spot workflow, the data source is the Tier-0. The T0AST Feeder is developed in order to poll regularly the Tier-0 database for a given workflow. This class inherits from the super-class Feeder, parent of all WMBS Feeders. T0AST Feeder injects new files once they appear in Tier-0 into the WMBS instance. Figure 6 shows the general design, using the Unified Modeling Language (UML)[13], of the FeederManager module.

Since the support of alignment and calibration workflows represents a high priority task for the CMS collaboration, the T0Emulator daemon has been developed in order to reduce the timescales to reach this goal. It allows to run the machinery over the beam spot determination workflow during down periods of the Tier-0. This daemon reproduces continuously data required for the beam spot workflow and which facilitates the development of the automation system.

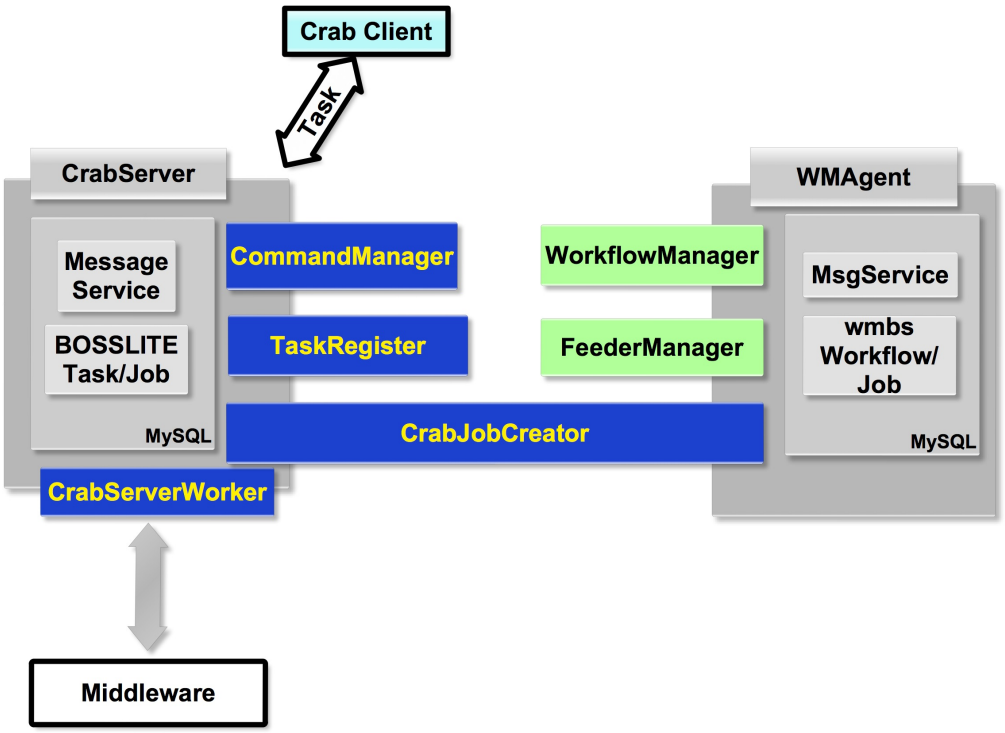

Figure 4: CRABSERVER and WMBS integration

```
datasetpath=/StreamExpress/Run2010A-TkAlMinBias-v4/ALCARECO

[WMBS]
split_value=10
feeder=T0AST
split_per_job=files_per_job
splitting_algorithm=FileBased
startrun=140251
automation=1

[USER]
return_data=1
```

Figure 5: WMBS section added in CRAB configuration file

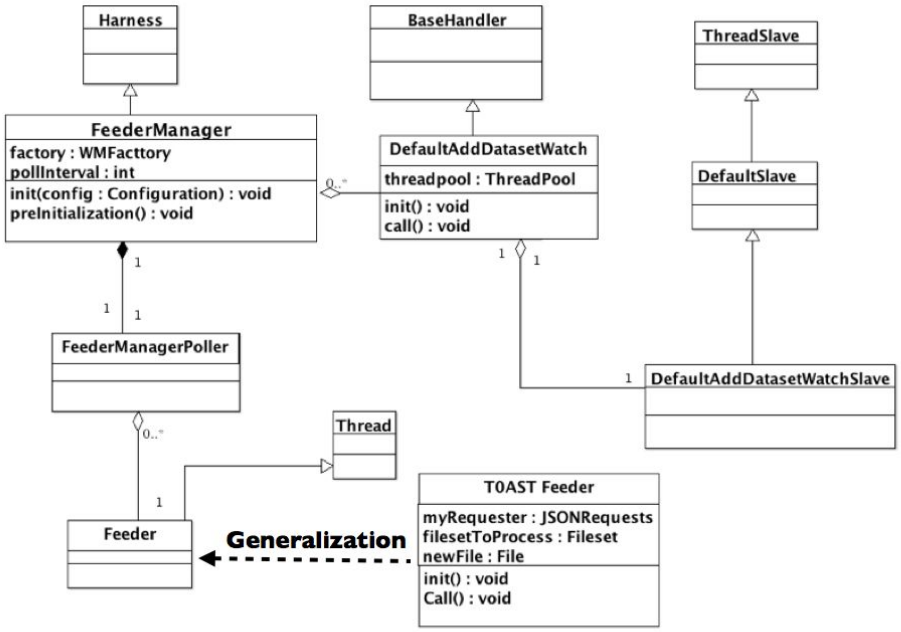

Figure 6: General Class diagram of the FeederManager

During the 2009 data-taking period, the beam spot was determined using manual scripts at a centre of mass energy of 900 GeV and this proved to be extremely valuable for checking the results given by the new code. During the down period of the accelerator, from the end of 2009 to the beginning of 2010, the alignment and calibration experts have validated the automation system in order to exploit it by the next LHC startup. For the beam spot calibration case, they used the system to run their workflow over old data that the reproduction was emulated using the T0Emulator. Since such data are already stored at the CAF when the LHC was running in 2009 and thus the beam spot values were already calculated, the validation step for this case was only the comparison of both results.

By the 2010 LHC startup, a dedicated production instance of CRABSERVER running the automation machinery has been setup to serve the alignment and calibration workflows. Monitoring the calibration of the beam spot workflow, the machinery takes less than 2 hours between the start of a run and the beginning of the automatic jobs submission over the data that is produced by this run. Figure 7 shows the fitted X positions of the beam line produced using the automation system as a function of luminosity sections for the run 132440.

Since the alignment and calibration workflows will run at CAF and thus will be limited by the Andrew File System (AFS) token lifetime, the lifetime of the workflows was extended to be as long as possible.

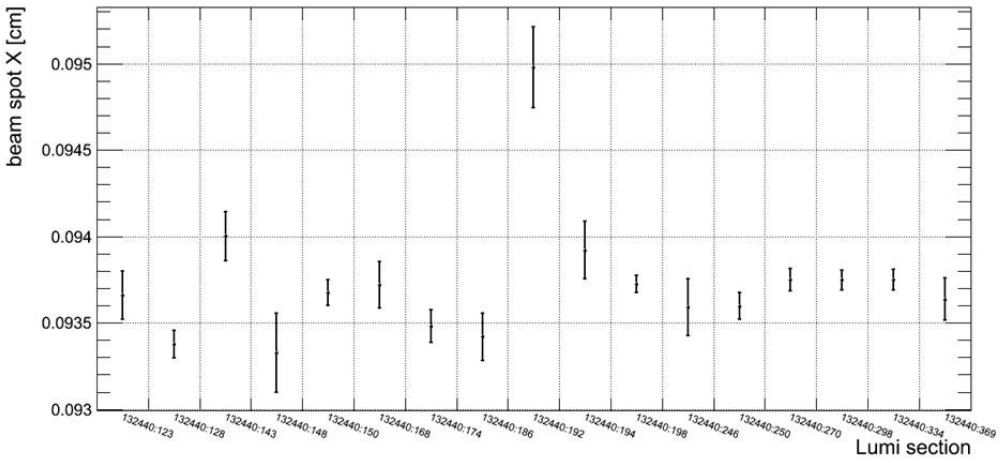

Figure 7: Fitted X positions of the beam line as a function of luminosity sections

6. Conclusions and long term strategy for automated workflows

CRABSERVER and WMBS library have been integrated in order to support automation of calibration workflows. The implementation described here is being tested by alignment and calibration experts in order to understand further requirements and to develop technical solutions.The integration has already been used with success to automate and tune the beam spot determination workflow.

The next effort will be spent to test tracker alignment workflows and drift tube calibration. Improvements of the CRAB interface for automated workflows are expected, based on feedback being received. Also, new features will be added to give alignment and calibration experts the ability to control and manipulate their workflows in a flexible way.

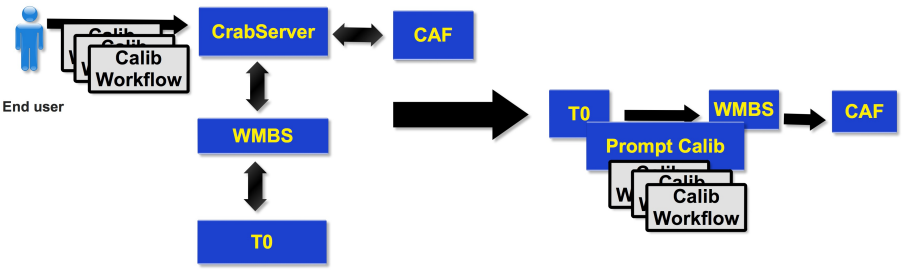

Figure 8: Long term CMS strategy

The strategy of CMS for alignment and calibration workflows is to use the system described here until workflows are stable i.e until they are fully tested and certified, following which they will

be migrated into the fully automated Tier-0 system. This is illustrated in Figure 8. In addition, this system will continue to be tuned and improved to automate typical CMS analysis workflows.

Acknowledgments

We wish to thank Lorenzo Uplegger and Francisco Yumiceva who shared with us their experiences using the automation system.

References

[1] CMS Collaboration, R. Adolphi *et al.*, *The CMS experiment at the CERN LHC*, *JINST*, 0803, S08004 (2008)

[2] CERN, http://cern.ch

[3] J. Spalding, Collaboration for two CMS working groups: the Machine Interface Group and Trigger Simulation Group, *Background Issues for CMS*, in *LHC Workshop on Experimental Conditions and Beam-Induced Detectors Backgrounds*, CERN, Geneva, Switzerland, 3 - 4 Apr 2008, pp.60-64

[4] S. Wakefield *et al.*, *Large Scale Job Management and Experience in Recent Data Challenges within the LHC CMS experiment*, in Proceedings of *XII Advanced Computing and Analysis Techniques in Physics Research*, Erice, Italy. Nov., 2008, PoS(ACAT08)032.

[5] D. Spiga *et al.*, *CRAB: an Application for Distributed Scientific Analysis in Grid Projects*, in proceedings of *the 2008 international conference on Grid computing - WorldComp*, GCA 2008, Las Vegas 14 - 17 July 2008, pp.187-193, ISBN: 1-60132-068-X

[6] D. Spiga *et al.*, *Automation of user analysis workflow in CMS*, in *17th International Conference on Computing in High Energy and Nuclear Physics*, Prague, Czech Republic, 21 - 27 Mar 2009

[7] G. Codispoti *et al.*, *Use of the gLite-WMS in CMS for production and analysis*, in *International Conference On Computing In High Energy Physics And Nuclear Physics*, 21 - 27 Mar 2009, Prague,Czech Republic, 15/05/2009

[8] S. Wakefield *et al.*, *Job Life Cycle Management libraries for CMS Workflow Management Projects*, in *17th International Conference on Computing in High Energy and Nuclear Physics*, Prague, Czech Republic, 21 - 27 Mar 2009

[9] A. Afaq *et al.*, *The CMS Dataset Bookkeeping System*, *Journal of Physics*, Conferences Series(119) 072001, 2008

[10] R. Egeland *et al.*, *Data Transfer Infrastructure for CMS Data Taking*, in Proceedings of Science, *XII Advanced Computing and Analysis Techniques in Physics Research*, Nov. 3 - 7, 2008, PoS(ACAT08)033.

[11] *XML Media Types, RFC 3023*, IETF. 2001-01. pp. 9-11. http://tools.ietf.org/html/rfc3023, Retrieved 2010-01-04.

[12] CMS Collaboration, *The CMS Computing Software and Analysis Challenge*, CMS IN-2008/044

[13] Craig Larman, *UML 2 et les Design Patterns*, Pearson Education, ISBN 2-7440-7090-4