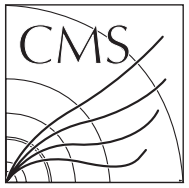
Available on CMS information server

CMS CR -2017/058



The Compact Muon Solenoid Experiment

Conference Report

Mailing address: CMS CERN, CH-1211 GENEVA 23, Switzerland

20 February 2017 (v2)

Continuous and fast calibration of the CMS experiment design of the automated workflows and operational experience.

P Oramus, G Cerminara, A Pfeiffer, G Franzoni, G Govi, M Musich and S Di Guida on behalf of the CMS Collaboration

Abstract

The exploitation of the full physics potential of the LHC experimentsrequires fast and efficient processing of the largest possible datasetwith the most refined understanding of the detector conditions. Toface this challenge, the CMS collaboration has setup an infrastructure for thecontinuous unattended computation of the alignment and calibrationconstants, allowing for a refined knowledge of the most time-criticalparameters already a few hours after the data have been saved to disk.This is the prompt calibration framework which, since the beginning ofthe LHC Run-I, enables the analysis and the High Level Trigger of theexperiment to consume the most up-to-date conditions optimizingthe performance of the physics objects. In Run-II this setup has beenfurther expanded to include even more complex calibration algorithmsrequiring higher statistics to reach the needed precision. Thisimposed the introduction of a new paradigm in the creation of thecalibration datasets for unattended workflows and opened the door to afurther step in performance.The presentation reviews the design of these automated calibrationworkflows, the operational experience in Run-II and the monitoringinfrastructure developed to ensure the reliability of the service.

Presented at *CHEP 2016 22nd International Conference on Computing in High Energy and Nuclear Physics*

Continuous and fast calibration of the CMS experiment: design of the automated workflows and operational experience

P Oramus[1,2], G Cerminara[2], A Pfeiffer[2], G Franzoni[2], G Govi[3], M Musich[4] and S Di Guida[5] on behalf of the CMS Collaboration

[1] AGH University of Science and Technology (PL)
[2] CERN
[3] Fermi National Accelerator Lab. (US)
[4] Universite Catholique de Louvain (UCL) (BE)
[5] Universita degli Studi Guglielmo Marconi (IT)

E-mail: piotr.oramus@cern.ch

Abstract. The exploitation of the full physics potential of the LHC experiments requires fast and efficient processing of the largest possible dataset with the most refined understanding of the detector conditions. To face this challenge, the CMS collaboration has setup an infrastructure for the continuous unattended computation of the alignment and calibration constants, allowing for a refined knowledge of the most time-critical parameters already a few hours after the data have been saved to disk. This is the prompt calibration framework which, since the beginning of the LHC Run-I, enables the analysis and the High Level Trigger of the experiment to consume the most up-to-date conditions optimizing the performance of the physics objects. In Run-II this setup has been further expanded to include even more complex calibration algorithms requiring higher statistics to reach the needed precision. This imposed the introduction of a new paradigm in the creation of the calibration datasets for unattended workflows and opened the door to a further step in performance. The presentation reviews the design of these automated calibration workflows, the operational experience in Run-II and the monitoring infrastructure developed to ensure the reliability of the service.

1. Introduction

CMS is a general-purpose detector at the CERN LHC. The CMS apparatus consists of many different layers, each designed to perform a specific task. In the finely segmented silicon sensors of the pixel and strip tracker charged particles can be tracked and their momentum measured. Precise determination of their energy is enabled by a lead tungstate crystal electromagnetic calorimeter followed by a brass and scintillator hadron calorimeter. The powerful coil of niobium-titanium superconductor bends the trajectories of charged particles, allowing their separation and momentum measurements. To identify muons, CMS uses three types of detector: drift tubes, cathode strip chambers and resistive plate chambers, all placed inside the magnet return yoke. Further description of the instrument is beyond of the scope of this document and can be found in [1].

The complexity of the system requires an elaborated framework for the management and computation of the detector calibration and alignment to make full use of its physics potential.

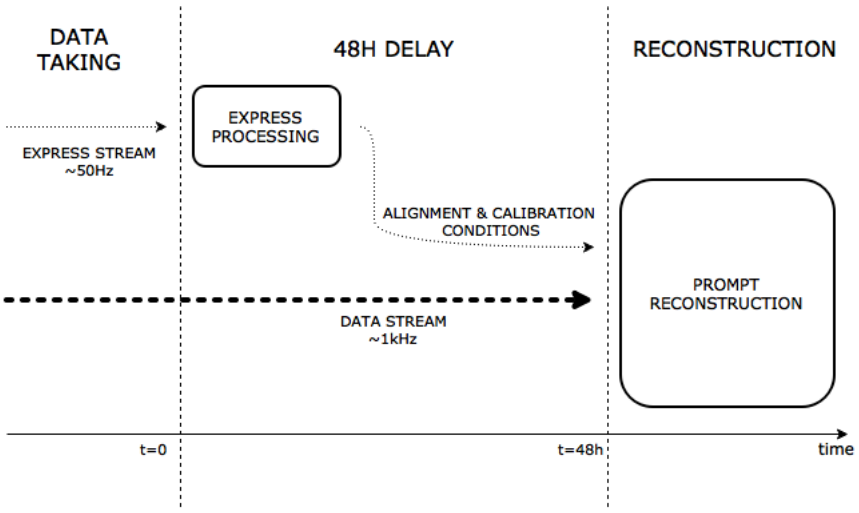


Figure 1. Schematic representation of the express and prompt processing of the data showing the latency of each step of the workflow chain.

One of the most crucial aspects in this context is the need of having the most accurate calibrations available with short turnaround. This allows for an efficient online event selection by the high-level trigger and delivery of datasets ready for physics analysis within few hours from their acquisition. The effort put into delivering this crucial feature resulted in the prompt calibration framework, running successfully since the beginning of LHC Run-I. In preparation for the LHC Run-II, new calibration workflows have been integrated in this framework; some of these required a new logic for the creation of calibration datasets to meet the needed statistical accuracy. After a short introduction about the prompt calibration framework, this paper will focus on the new functionality introduced to cope with the new use cases.

2. Organization of the data reconstruction process

The organization of the whole process of the first reconstruction pass is dictated by the goal of having a set of accurate calibrations available in a very short time. Running the low-latency calibration workflows is possible as a result of the 48 hours delay between data acquisition and the actual reconstruction of the physics objects. Right after the data are collected, a dedicated stream, called express stream, is first processed, providing a quick feedback about the detector status, physics performance and yielding the input for the calibration workflows. These results are usually available one or two hours after collecting the raw data. During regular working conditions of the CMS experiment, the express stream data rate oscillates around 50 Hz while the main data stream carries about 1 kHz of data. This mechanism is illustrated by Figure 1.

2.1. Workflows running on the Tier0 computing farm

In the current setup exercised over the LHC Run-I and ongoing Run-II, the automated calibration workflows are executed for each acquisition run. These can last from less than one to over a dozen hours depending on the LHC and the detector operational status. Figure 2 shows the steps performed to derive the calibrations.

The express data belonging to the same run are processed by parallel jobs using the CMSSW reconstruction package on the Tier0 computing farm. Both the reconstruction of the physics objects for further analysis and the selection of the events for the alignment and calibration algorithms are executed during this step. The events selected as input to the calibration

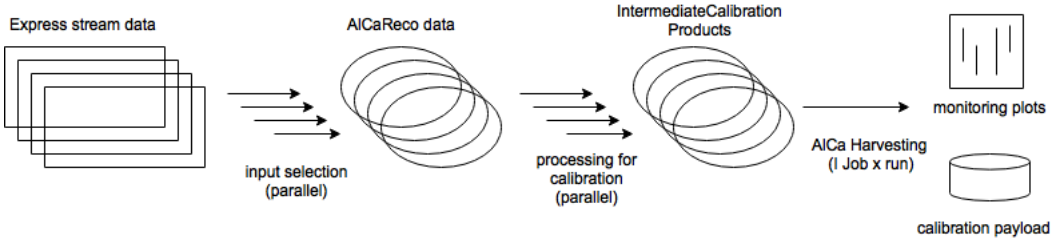


Figure 2. Schematic view of the automated alignment and calibration workflows for one run on the Tier0 computing farm.

algorithms are stored in special datasets called *AlCaReco*. This step is usually the most CPU intensive of the whole workflow. The job splitting ensures that each application receives chunks of contiguous events keeping as the atomic quantum of the data, called luminosity-section, corresponding to about 23 s of data-taking. The AlCaReco dataset is further processed to compute the intermediate calibration products and can be either histograms or calibration constants. Finally, the intermediate products are aggregated for a given run into a set of conditions as an input for future reconstruction operations. This step, called *AlCa Harvesting*, is computed by a single job per run and produces as output one or more database payloads. The payload is first stored in a SQLite file and then transferred to a Oracle database storing the Alignment and Calibration constants [2]. Furthermore, histograms in ROOT [3] format are also produced and injected to an instance of the Data Quality Monitoring (DQM) [4] framework. By using the web-based Graphical User Interface provided by DQM, detector experts can review the histograms inspecting the performance of the calibration algorithms.

2.2. Alignment and calibration workflows

Several time-critical calibrations have already been exercised in the automated calibration system during the LHC Run-I and more have been added in preparation to the Run-II. Currently the following workflows are routinely used in production:

- fit of the luminous region position and width (using tracks);
- identification of silicon strip tracker transient problematic channels for event-by-event optimization of the pattern recognition reconstructing the trajectories of charged particles;
- determination of charge gains of the silicon strip tracker;
- track-based alignment of silicon pixel inner tracker mechanical supports structures.

A more detailed explanation of these calibration algorithms goes beyond the scope of this report. A description of the performance and operational experience of the workflows can be found in [5].

3. Multi-run AlCa Harvesting

In the LHC Run-II the prompt calibration framework setup has been further expanded to include even more complex calibration algorithms requiring higher statistics to reach the needed precision. For this purpose, a new paradigm in the creation of the calibration datasets for unattended workflows has been introduced. The idea standing behind the new approach, called *multi-run AlCa Harvesting*, is based on running the last step of the workflow previously described on a broader set of input data consisting of multiple runs. This section focuses on this particular development.

In the interest of data homogeneity, calibrations performed on more than one run need to use consistent data as input. For this reason, a few properties are checked to be consistent with the alignment and calibration algorithms needs:

- only data belonging to the same dataset era are aggregated: changes in the dataset era usually mark crucial differences in the trigger configuration, CMS running conditions or LHC parameters;
- the input data need to be acquired with the same magnetic field;
- cosmic and collision data are treated separately;
- the input data need to be reconstructed with the same version of the reconstruction software and architecture;
- consistent set of alignment and calibration conditions used for the reconstruction.

Each multi-run component has to preserve similar features in order to produce reasonable results. Data holding a defined set of properties can belong to exactly one multi-run dataset, preventing the execution of the mechanism multiple times on the same input. Multi-run input datasets are formed trying to meet the minimal requirements to achieve the needed accuracy without compromising on the time granularity of the calibrations. Distinct multi-run datasets have to be fully separate in terms of content, thus allowing the computation to run in parallel without resource conflicts.

3.1. Data discovery and multi-run input dataset creation

The procedure of multi-runs creation is initiated by the data discovery process. First, the Run Registry service [4] is queried for the recent CMS runs distinguishing those acquired during the LHC collisions from those acquired during cosmic data-taking. For each of these new runs, the Dataset Bookkeeping System (DBS) [6] is used to discover the files and datasets produced by the express workflows described in Section 2.2. Furthermore, a dedicated web based API is used to retrieve the configuration of the application running on the Tier0 computing farm.

The multi-run input dataset is assembled out of the runs with similar properties and extended with new data as long as it is needed for the algorithms to produce the set of calibration constants. The most basic condition which triggers the multi-run AlCa harvesting processing, is based on the total number of events aggregated by the multi-run dataset. For every single calibration workflow, the events threshold is determined experimentally by using the generate and test method - whenever for given condition the computation did not produce expected payload, the term is increased, with the aim of finding the optimal compromise between statistical accuracy and time granularity of the calibration.

Subsequent items can only be formed after the processing is successful, i.e. when the algorithms delivered the set of alignment and calibration constants.

The data discovery algorithm needs also to account for the different processing time of the various runs, mostly dependent on their length and the luminosity of the LHC fill. For this reason the system queries the Tier0 API to know the status of the processing for each of the candidate runs and making sure that only those runs for which the express processing is finished are added in time order to the multi-run dataset.

3.2. The state machine for multi-run dataset processing

The data discovery, the processing of the multi-run datasets and the handling of the calibration and DQM products have been designed to be fully independent, allowing the implementation of a state machine embracing the whole process. The state machine introduce great resilience by:

- allowing for independent and parallel handling of multi-run datasets without the need of scheduling a priori the various steps;

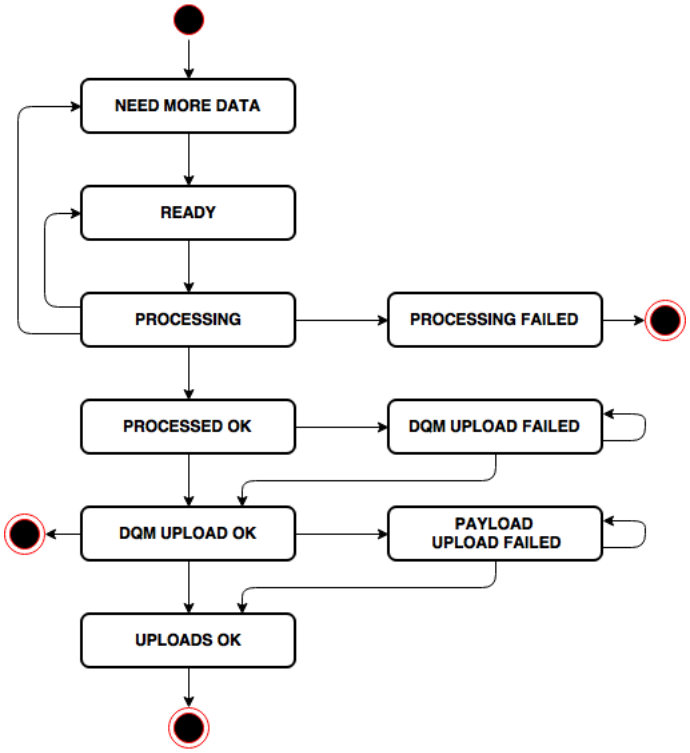


Figure 3. The state machine diagram illustrating the complete set of possible transitions.

- allowing for resubmission of problematic steps of the processing since the status of multi-run dataset remains well defined at any moment in time;
- simplifying the monitoring of the workflows.

Each multi-run dataset is assigned one of nine possible states:

- *Need more data* - waiting for more events to satisfy the algorithms requirements;
- *Ready* - ready to be picked up by an operation executor;
- *Processing* - in state of performing AlCa Harvesting computation which can last from few minutes up to few hours;
- *Processing failed* - needs human intervention as the processing failed too many times;
- *Processed ok* - processing finished successfully, expecting histograms upload to the DQM framework instance;
- *DQM upload failed* - the DQM upload failed and needs to be repeated;
- *DQM upload ok* - an end state for workflows not requiring the payload upload;
- *Payload upload failed* - the payload upload failed and needs to be repeated;
- *Uploads ok* - an end state indicating a thorough success.

The decision to change the multi-run status is based on the result of a single step, usually by analyzing a job output and an exit code.

One of the most important goals of the system is to provide unattended computation of alignment and calibration workflows. For this reason, a special attention was paid to the error handling. The processing and upload steps can be automatically retried a fixed number of times, set by the system administrator. Whenever the given failures threshold is reached implying a problem, qualified experts are notified via an e-mail that a manual intervention is required. This information is also displayed on the monitoring web page described in the following.

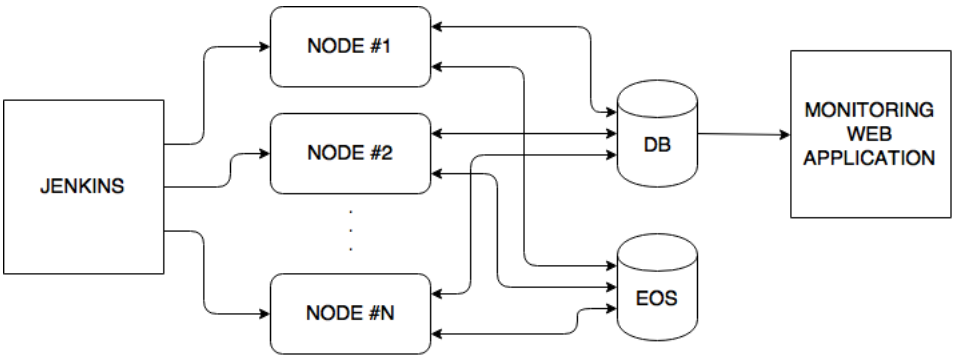


Figure 4. Architecture of the multi-run AlCa Harvesting multi-run system.

3.3. Multi-run harvesting: system architecture

The CERN Tier0 computing farm runs all the alignment and calibration workflows, including the CPU-intensive ones. The multi-run AlCa harvesting system is designed to process only the last harvesting step. For this reason, the data discovery, multi-run AlCa harvesting and upload steps are all executed on a dedicated nodes allocated to CERN Analysis Facility (CAF) machines, with a minimal environment prepared on top. These nodes are dedicated to calibration and alignment computing and meet all the requirements for running the processing.

An important role in the system is played by a Jenkins server, whose main function is to orchestrate the tasks by spawning them in a predefined time scale. It is also responsible for alerting users in case of errors, storing the system configuration and passwords in encrypted state, connecting to remote nodes. When creating a job, Jenkins connects to one of the available nodes using an interactive batch queue, each time cloning needed source code from the Git repository and creating a Python virtual environment. The most important files are backed up in a disk space provided by the EOS [7] storage solution to allow to reproduce the job in case of need. Afterwards the whole workspace is cleaned, helping to keep the environment in a sterile and intact state. All the intermediate results of computation and the state of the each multi-run dataset are stored in an Oracle database, which is a central part of the system accessible from every involved node. The database stores entire information concerning multi-run datasets allowing the processes to communicate with each other. A part of this data is then utilized by the web application, created for user-friendly process monitoring, as being depicted in the following paragraph. Figure 4 illustrates the overview of discussed architecture in a more general way.

4. Monitoring infrastructure

All the services described in this report are meant to run in an unattended way, meaning that the human factor should be minimized as much as possible. To support this goal, a monitoring layer has been provided, feeding a user with valuable information about utility status and being capable of issuing alarms in the event of problems. The subsequent sections focus on outlining two of the applications, which were designed to present the reports through an easily accessible web interface.

4.1. pclMon: an application for monitoring of unattended automatic workflows running on the Tier-0 farm

Since LHC Run-I a web-based application called pclMon exists, showing high efficiency in coping with unexpected complications during the unattended computation of automatic workflows. The service aggregates information from various sources with the goal of promptly identifying possible

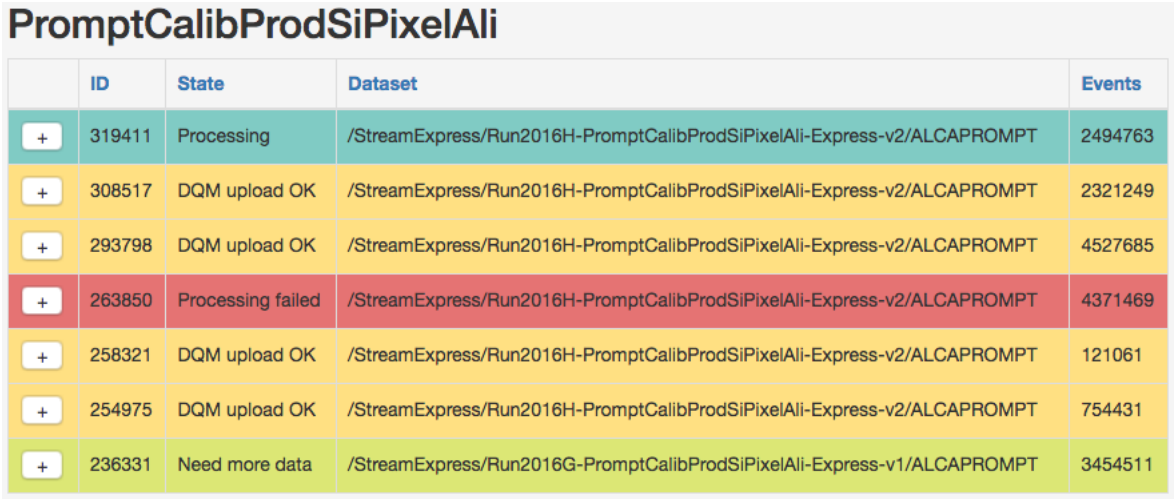


Figure 5. Screenshot of the monitoring web application showing a few multi-runs along with their state and basic properties.

issues which might delay the computation of calibration sets. It continuously monitors all the steps of the automated workflows running on the Tier0 computing farm. For each calibration workflow it displays properties such as:

- start and end time of the run to be calibrated;
- status of the Tier0 processing (and in particular of the AlCa Harvesting step);
- runs for which the minimal requirements in terms of statistics are not met;
- status of uploads;
- status of the advancement of the bulk processing for prompt reconstruction on the Tier0 computing farm.

Whenever one of these quantities displays a problem, potentially impacting the availability of the updated calibration conditions in time for the prompt processing of the bulk of the data, an alarm is issued both via e-mail and on the application front-end.

4.2. Monitoring of multi-run based workflows

For the multi-run AlCa Harvesting framework a new monitoring web application has been developed. The application offers an overview of the system by displaying detailed properties of individual calibration job and thus helping with further investigation when needed. Using the interface user can directly access linked DQM histograms, logs of the calibration payload upload or even Jenkins records for a specific job. A significant emphasis was put on showing the multi-run state, which was primarily aimed at highlighting problematic events.

In order to profit from full capabilities of different technologies, the service implementation was divided into two parts. The first one, written in Flask, exploits the content of the multi-run database exposing prepared data through a REST API. The JavaScript-based front-end on the other hand is designed to present obtained information to the user.

5. Conclusions

The demand for the rapid delivery of alignment and calibration constants for the CMS experiment operations led to the setup of a powerful prompt calibration framework. This framework has been successfully exploited since the first LHC runs, demonstrating the needed

reliability and flexibility to run several kinds of calibration workflows. In the ongoing LHC run, new uses cases arose which required to expand the functionality of this framework to cope with calibrations needing higher statistics then available on a single CMS acquisition run. The new dataset creation paradigm allowed more advanced calibration algorithms to operate on a new, hitherto unexplored level.

References

[1] S. Chatrchyan *et al.* [CMS Collaboration], The CMS Experiment at the CERN LHC, JINST **3** (2008) S08004.

[2] S. D. Guida, G. Govi, M. Ojeda, A. Pfeiffer and R. Sipos, The CMS Condition Database System, J. Phys. Conf. Ser. **664** (2015) no.4, 042024.

[3] I. Antcheva *et al.*, ROOT: A C++ framework for petabyte data storage, statistical analysis and visualization, Comput. Phys. Commun. **180** (2009) 2499.

[4] F. De Guio [CMS Collaboration], The CMS data quality monitoring software: experience and future prospects, J. Phys. Conf. Ser. **513** (2014) 032024.

[5] G. Cerminara and B. van Besien, Automated workflows for critical time-dependent calibrations at the CMS experiment, J. Phys. Conf. Ser. **664** (2015) no.7, 072009.

[6] M. Giffels, Y. Guo, V. Kuznetsov, N. Magini and T. Wildish, The CMS Data Management System, J. Phys. Conf. Ser. **513** (2014) 042052.

[7] A. Peters, E. Sindrilaru and G. Adde, EOS as the present and future solution for data storage at CERN, J. Phys. Conf. Ser. **664** (2015) no.4, 042042.