

CMS ECAL DAQ Monitoring system

Giacomo Cucciati^{1,*}, on behalf of the CMS Collaboration

¹CERN, CH-1211 Geneva 23, Switzerland

Abstract. The Large Hadron Collider (LHC) at CERN in Geneva, Switzerland, has just completed the Run 2 era, colliding protons at a center-of-mass energy of 13 TeV at high instantaneous luminosity. The Compact Muon Solenoid (CMS) is a general-purpose particle detector experiment at the LHC. The CMS electromagnetic calorimeter (ECAL) has been designed to achieve excellent energy and position resolution for electrons and photons. A multi-machine distributed software configures the on-detector and off-detector electronic boards composing the ECAL data acquisition (DAQ) system and follows the life cycle of the acquisition process. Since the beginning of Run 2 in 2015, many improvements to the ECAL DAQ have been implemented to reduce and mitigate occasional errors in the front-end electronics and not only. Efforts at the software level have been made to introduce automatic recovery in case of errors. Automatic actions has made even more important the online monitoring of the DAQ boards status. For this purpose a new web application, EcalView, has been developed. It runs on a light Node.js JavaScript server framework. It is composed of several routines that cyclically collect the status of the electronics. It display the information when web requests are launched by client side graphical interfaces. For each board, detailed information can be loaded and presented in specific pages if requested by the expert. Server side routines store information regarding electronics errors in a SQLite database in order to perform offline analysis about the long term status of the boards.

1 Introduction: the electromagnetic calorimeter

The Compact Muon Solenoid (CMS) is a multipurpose detector designed for the precision measurement of leptons, photons, and jets, among other physics objects, in proton-proton as well as heavy ion collisions at the LHC [1]. It is placed in one of the crossing points, and up to 70 inelastic interactions happen inside the detector at every particle bunch-crossing (BX). The electromagnetic calorimeter (ECAL) is the CMS sub-detector designed to measure the energies of electrons and photons [2]. ECAL is composed by three major sections: ECAL barrel (EB), ECAL endcap (EE), and endcap preshower (ES). EB, in the shape of a cylinder, comprises 61200 $PbWO_4$ scintillating crystals while EE includes 14648 crystals. ES is composed of 4288 silicon sensors of 32 strips each and it is placed in front of EE. The higher granularity can distinguish pairs of photons produced by π^0 decay that can be wrongly identified as single photons by the larger section of the crystals. There are separate DAQ systems for the EE+EB (called "ECAL" in the DAQ) and ES detectors. The DAQ systems of ECAL and ES are made of an on-detector part placed close to the detectors and by an off-detector part [3], situated in the service cavern (USC) of CMS (Figure 1).

*e-mail: giacomo.cucciati@cern.ch

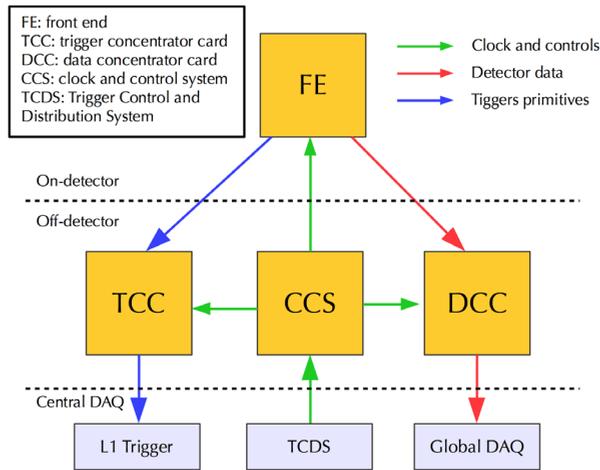


Figure 1. ECAL DAQ system schematics with Trigger, Control and Data paths highlighted.

2 ECAL/ES DAQ online software

The main task of the DAQ online software is the control of the readout electronics. Specifically, it configures the electronics (off and on-detector) with the parameters required to tune the data taking. It constantly monitors the status of the hardware, reporting errors to the operator. It sends control commands to the Finite State Machine (FSM) of the boards' firmware. It integrates the ECAL sub-system in the CMS Trigger and DAQ system (TriDAS) [4]. The communication between the computers where the software is deployed and the VME crates hosting the off-detector electronic boards is granted by specific CAEN modules connected by mean of optical fibers. In particular a A3818 PCIe Controller is mounted on the machines and a V2718 VME Controller (or Optical Link Bridge) is placed in the first slot of each VME crate. ECAL and ES use in total about 15 machines to control the hardware. In order to support this modular distributed system, a custom library has been developed in the CMS collaboration, called XDAQ [5]. Some functionalities are: inter-process communication with SOAP-based channels, vendor independent database access service, monitoring and error-reporting infrastructure, a hardware access library (HAL) for VMEbus communications, a FSM to control the phases of the data acquisition. The HAL library can instantiate a VMEDevice object that represents virtually an electronic card and it can easily perform read and write operations on the hardware registers (Figure 2).

2.1 Function Manager

The online software follows a hierarchical organization. At the top of the hierarchy are the Function Managers (FM), web applications based on the Run Control and Monitoring System (RCMS) Java framework. They offer an interface for the operator to manage the life-cycle of the lower layer applications. A dedicated FM is instantiated for each sub-detector. They can act in stand-alone mode, but generally they are controlled by a common Level-0 FM, that, above all, represents the control panel of a CMS data acquisition run. Via the Level-0 function manager, the operator can set the configurations required by the data taking, add/remove the sub-detectors to/from the run, control the phases of data acquisition (Configure, Start, Stop...). Moreover all the problems and error conditions in the sub-systems are delivered to

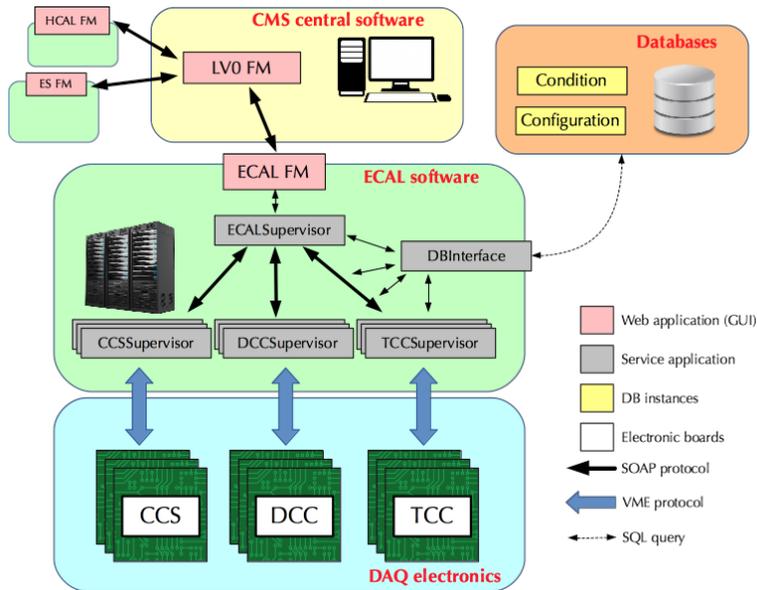


Figure 2. Schema of ECAL DAQ control software. ES follows the same schema with Supervisor applications dedicated to its electronics.

the highest FM for a prompt operator response. All the operations performed by the user at Level-0 are propagated through to the lower layers, first the sub-detector FM and later to the single supervisor applications.

2.2 Supervisor applications

The building blocks of the ECAL DAQ software are the ECAL Supervisor and the Resource Supervisors, C++ class instances inheriting from the common generic XDAQ class. The ECAL Supervisor is responsible for the coordination of the different DAQ resources in the sub-detector: it receives commands from the ECAL FM and dispatches them to Resource Supervisors. A Resource Supervisor is launched for each component of the off-detector electronics and it instantiates a VMEDevice object to communicate with the hardware. The Supervisor extracts the configuration parameters from an SQL database according to the directives coming through the ECAL Supervisor and it writes them on the registers of the board. Every 5-10 s, each application contacts the hardware and retrieves the status as well as other information on the ongoing run.

3 Electronics errors handling

3.1 FE errors and SEU

The on-detector electronics are built to be radiation tolerant, but occasionally high energy particles crossing the boards induce errors. If the passage of a particle flips one of the bits of the words stored in the FE buffers, the information carried by that fragment is modified. The data are nonetheless sent to the Data Concentrator Card (DCC) and subsequently to the central DAQ farm. However, the event will be rejected in the offline analysis if the new word

Table 1. Comparison of the collision data loss in fb^{-1} , for ECAL and ES during the 2016 and 2017 data taking periods.

Year	LHC delivered (fb^{-1})	CMS recorded (fb^{-1})	EB/EE + ES losses (pb^{-1})
2016	41.07	37.82	237.12 + 189.23 (0.6% + 0.5%)
2017	50.25	45.39	151.57 + 45.12 (0.3% + 0.09%)

generated has no meaning. There are specific cases called Single Event Upset (SEU) where the modification of the word induced by the radiation can completely break the data integrity and the DCC is not able to produce the event fragment. The consequence is an error in the electronic board and the stop of the data flow from the sector associated with the affected DCC. The Timing and Control Distribution System (TCDS) will stop the Level-1 Accept flow and the run is blocked. By means of the Level-0 FM, the operator can restart the data taking by only reconfiguring the ECAL or ES system again. The entire procedure requires at least 5 minutes since the applications have to be killed and re-initialized while electronics have to be reconfigured with parameters obtained from the database. Since data acquisition is stopped, this delay is considered CMS downtime. To avoid these unpredictable interruptions a recovery mechanism has been implemented. The error caused in the board by a SEU can be detected by the Resource Supervisor and communicated to the RCMS Level-0 that pauses the run (triggers are stopped). At this point the software that has detected the problem can fix the error by applying a reconfiguration of the FE electronics. When the recovery action is concluded, the Level-0 is informed and the triggers are enabled again, resuming the run. The main advantage of this architecture is the reduction of the luminosity loss, since a blocking error is solved with an on-the-fly reconfiguration that requires only 20-40 s.

3.2 Software improvements

In 2017, the ECAL DAQ group has started a campaign to reduce the downtime caused by electronics errors. A first strategy was based on an extension of the SEU recover mechanism to some of the off-detector electronics. The main downtime source has been found in clock perturbation inducing errors in the ES DCC boards. The ES-DCC Supervisor monitors the boards every 5 s and if 2 or more fibers are reporting error, a “Recover SEU” request is sent. The action sequence is similar to the one adopted for SEUs, the only difference is that the DCC for an off-detector board, rather than an on-detector boards, is reconfigured. The ECAL/ES DAQ software has been also improved to react to resets of the FE electronics induced by radiation or power glitches. The same monitoring action used by the Clock and Control System (CCS) Supervisor to intercept SEUs has been upgraded to recognize this kind of blocking error and reacts by launching a SEU recovery procedure. The upgrades applied to the software have brought a clear improvement in the reliability and efficiency of the ECAL and ES DAQ system. The percentage of luminosity loss/luminosity delivered for ECAL and ES DAQ system has been reduced respectively from 0.6% and 0.5% in 2016 to 0.3% and 0.09% in 2017 (Table 1).

3.3 Auto-masking procedure

Between 2017 and 2018, the ECAL and ES code has been further upgraded to automatically recognize and mask broken channels. Some occasional FE electronics errors cannot be fixed even if a SEU recovery mechanism is applied. Even if these incidents are very rare, they can be potentially dangerous for the data taking since the source of the problem requires time to

be identified and isolated by the DAQ experts. With the new release, if the issue is confined to a few FE channels, the software masks them for the ongoing run. The CCS Supervisor reconfigures only the recoverable channels and sends the list of broken channels to the DCC Supervisor via SOAP message. The latter will reconfigure the DCC board, disabling the data readout from the corresponding fibers and thus avoiding the block of the run due to possible data integrity errors. This procedure allows DAQ experts to investigate the problem while the run is still ongoing, reducing the downtime induced by the issue.

4 EcalView, the ECAL/ES monitoring system

The Resource Supervisors report the information that are read periodically from the hardware in log files, but it is often difficult for the operator to extract an understanding of the situation from the texts. Moreover hundreds of parameters can be read from the electronics and they must be displayed in a user-friendly and easy-to-read format. After the recent software improvement, it has become even more important to have a clear picture of electronics status, such as the SEUs that have occurred and the channels automasked by the software. Therefore it has been necessary to build a monitoring system that is able to collect the information distributed over several machines and to display them clearly for experts. The first step is already performed by a XDAQ web application called Slash2g, provided by the CMS central online system. Running on a machine accessible by the VME PCs, the application is able to store information in virtual tables (or flashlists). A dedicated flashlist is instantiated for each board type and each Resources Supervisor has a dedicated row inside the table to write the data. For instance the flashlist used by the DCC Supervisors has 54 rows, one per DCC board. At each reading, the Supervisor publishes the electronics status in the table, overwriting the previous values contained in the row, and the table will offer a real time picture of the situation (although in reality, the information can be 5-10 s old). The flashlists are published on the CMS net and they can be visualized by the operator directly through a web browser. However, the great number of parameters contained by each row makes the flashlist a complicated tool. This problem is solved by the new application EcalView. EcalView is divided into two parts: the back-end whose objective is to read the flashlists and elaborate the data to create a summarized view of the electronic status; the front end, used to present the data to the user in a clear way (Figure 3).

4.1 Back-end: server side

On server side, the data flow starts with the Flashlist Pollers. These simple routines periodically contact the Slash2g application and retrieve the tables in JavaScript Object Notation (JSON) format. A Poller is instantiated for each flashlist published by Slash2g. At each reading, the Poller broadcasts an event "new-raw-data" (available). The second component of the chain is composed by several Monitor applications. A Monitor is subscribed to one or more flashlists and a specific callback function is associated with the "new-raw-data" event that it will receive. The function takes the raw data from the flashlist and transforms them into a summary of the electronic status. The Monitor also has two other roles: storing the electronics errors in a database in order to create a history of the ECAL status; providing data and pages in response to browser requests. For this reason each monitor is generally composed of three elements: the Monitor Logic for data elaboration, the Database Monitor Interface (DMI) for SQL actions, and the Monitor Interface for browser responses. Through the Monitor Interface, the Monitor Logic will broadcast an event "new-data" (available) every time it has finished to elaborate the new information obtained from the Flashlist Poller. The time required to elaborate the data is less than the poller period so the monitor pace is

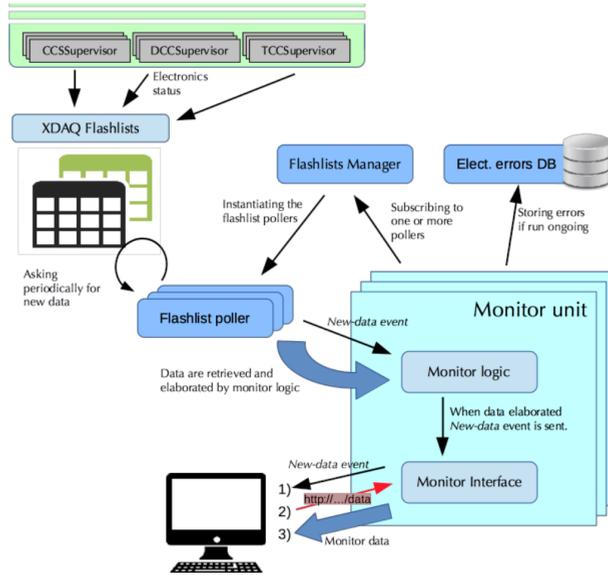


Figure 3. Schematic representation of the EcalView application.

completely driven by the latter. The application is based on the JavaScript engine Node.js v6.12.3 extended with Express, a framework to create web applications and to interface the functions with user requests. This technology has been chosen because it can easily instantiate a light server with an asynchronous JavaScript engine (not multiprocessing). The engine fits the requirements of EcalView, an application with low CPU duty and long idle periods. Moreover the back-end and front end use the same programming language, thus facilitating the development. The database chosen is provided by SQLite and a table dedicated to each Monitor has been instantiated.

4.2 Front end: client side

The user has access to the graphical interface via web browser. The interface, or front end, presents to the user an access for each Monitor main page. Generally the main page is composed of a matrix of boxes, each one displaying the status of a single board. For instance the DCC Monitor shows a matrix of 9×6 boxes representing the 54 DCC boards. Just below the matrix, the Monitor page shows a table with the errors of the electronics currently present during the run. At the bottom of the page a filtered table with the full history of the errors has been inserted as well (Figure 4). In some of the Monitors the user can access a secondary page containing detailed information about the status of a board, simply clicking on the corresponding box on the main page. When the main page of a Monitor is opened, a WebSocket channel is created between the browser page and the Monitor Logic on the server. In this way the client side can receive a "new-data" event, and consequently via AJAX request it can retrieve the new summary generated by the Monitor Logic. The response will also contain the list of the current errors. With this method the Monitor will be automatically refreshed with the new information only when they are effectually available on the server side, limiting the number of requests and the data traffic. A similar process is used for the error history in which the table is updated only when an event "history-updated" is received by the client monitor. To generate the web pages, the framework Vue.js and the module bundler Webpack



Figure 4. Main page of the TCC monitor showing the TCC boards table, the current error table and the error history.

have been used. These libraries have been chosen mainly because they offer a modular implementation of web pages via templates and they come with a simple linkage between data displayed and back-end information, with an auto-refresh on data change. In order to monitor software decisions as auto-recovery or auto-masking, it has been decided to apply the same mechanism used for electronics errors. The DCC Supervisors publish the detail of the events (SEUs and channels masked) on the flashlist and the Monitor Logic can retrieve them with two consequences: they are saved in the database for future tracking and they are directly displayed on the error tables.

4.3 Warnings and service monitors

Any ECAL user can check the ECAL electronics errors by scrolling through the Monitors. But some errors require a prompt action by the ECAL DAQ expert since they can cause serious data taking problems, such as blocking the run or sending bad data for extended areas of the detector. In order to trigger experts attention, some warning are sent by the application.

In particular two modules have been added to communicate problems via email and on an open source messaging platform, where an internal channel is used by the ECAL group. The Monitor-Logic can have access to these features simply importing the modules. Every time new data are retrieved, the Monitor-Logic elaborates them and in case of serious errors, it prepares the message to be communicated externally.

EcalView has been recently updated to also provide service monitors. They are not connected to an ECAL electronics flashlists but they offer auxiliary information like the statistics of errors per FED or the event fragment payload generated by each supermodule during the run.

4.4 Performances

EcalView has been deployed on a 8 GB RAM virtual machine with eight 2.3 GHz CPUs. The virtual machine runs in the dedicated CMS network (closed to the global web). The process uses on average 11.5% of CPU time and 175 MB (2.2%) of the system memory. The low resource consumption of the application is due to the long idle periods as expected from the cycle time of the Flashlist Pollers. The payload of the client requests depends on the Monitor displayed by the browser. The average rate is 15 kB/s with a peak of 100 kB/s for the TCC monitor. The entire project requires 346 MB of disk space, excluding the database files. In one year of activity, the database has stored 140 MB of information even if the increasing rate clearly depends on ECAL status and activities.

5 Conclusions

During 2017, ECAL and ES have shown an important reduction of the downtime due to DAQ system problems, with only the 0.3% and 0.09% of the luminosity loss with respect to the amount delivered by LHC. The improvement is due to software upgrades focused on the recognition of on-detector and off-detector electronics errors and the application of a hardware recovery procedure. The upgrades have also introduced an auto-masking action of the unrecoverable channels applied with just a short pause of the CMS data taking. A new web application, EcalView, has been released to display the status of the ECAL and ES electronics in real time. It reports the errors detected by the hardware and the automatic actions taken by the software, sending warnings directly to the experts in case of critical situations. The application is based on Node.js JavaScript engine on the server side and Vue.js on the client side. EcalView is light and stable, as well as easily expandable due to its modular structure.

References

- [1] CMS Collaboration, JINST **3** (2008).
- [2] CMS Collaboration, CERN-LHCC **97-033** (1997)
- [3] R. Alemany et al., IEEE Trans. Nucl. Sci. **52**, 1918–1924 (2005).
- [4] P. Musella, CERN-THESIS-2010-240 (2010).
- [5] L. Orsini at al., Online documentation, <https://svnweb.cern.ch/trac/cmsos>