

DQM4HEP : A generic Data Quality Monitoring for High Energy Physics

Rémi Été

Univ, Lyon, Université Lyon 1,
CNRS/IN2P3, IPNL 4 rue E Fermi
69622, Villeurbanne CEDEX, France
Email: rete@ipnl.in2p3.fr

Antoine Pingault

Ghent University, Department of Physics
and Astronomy Proeftuinstraat 86,
B-9000 Gent, Belgium
Email: antoine.pingault@ugent.be

Laurent Mirabito

Univ, Lyon, Université Lyon 1,
CNRS/IN2P3, IPNL 4 rue E Fermi
69622, Villeurbanne CEDEX, France
Email: mirabito@ipnl.in2p3.fr

I. INTRODUCTION

With increasingly sophisticated devices, online data quality monitoring is of a significant importance for the detector and operation efficiency. Monitoring data is also a first step to a certification of the reliability of the recorded data for off-line physics analyses. Experiments usually develop their monitoring system on top of their event data model and file format to access and store data. This dependency has the tendency to tie the monitoring system to its dedicated experiment and leads to difficulties to adapt it to others.

With this in mind, a generic online Data Quality Monitoring (DQM) system has been developed without any assumption on the event data model (EDM) and data type to treat. A dedicated implementation, based on the LCIO [6] event data model for the Linear Collider collaboration was also developed. This implementation will be put to real condition testing using a combined detector setup with the CALICE ¹ Semi-Digital Hadronic CALorimeter (SDHCAL) and Silicon Tungsten Electronic Calorimeter (SiWECal) prototypes during test beam campaigns at CERN².

II. KEY POINTS AND SOFTWARE ARCHITECTURE

The framework is strongly based on a plugin system that allows some part of the software to be switched from one to another. As a consequence, the data serialization used to send data across the network is encapsulated in plugins, loaded at run-time. It is also designed to be run in separate processes linked to each other by TCP/IP or HTTP protocols using DIM[2] and Mongoose[3] respectively.

Fig. 1 shows the monitoring framework data flow from incoming data sources to the client visualization computers. An important effort has been put on the creation of a generic interface to the data acquisition system and the development of an event builder. An independent data acquisition (DAQ) process dumps the data sources content in the shared memory (shm) until a given limit in bytes is reached (set by the DAQ). In this way, a slow data treatment by the monitoring will not affect the data taking and writing to disk. Since the data structure is setup dependent, the implementation of the event

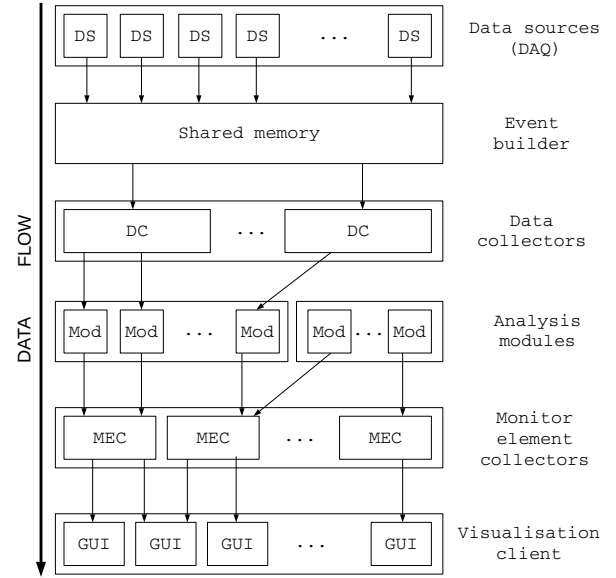


Fig. 1: The monitoring framework data flow: from data sources to visualization.

builder becomes highly specific. The event building is encapsulated in series of *shm processors* that convert data sources buffers into the user data structure. The whole reconstructed event is then serialized, dispatched across the network and stored into one or multiple data collector server applications.

Client interfaces are provided to either perform manual queries of collected data or work in update mode for which the data is directly forwarded on reception. User's online data analysis are also implemented as plugins in the system and steered using configuration files, increasing the modularity of the framework. They use a data collector client interface in update mode to receive and process data.

The main aim of the online data analysis is to reduce the initial amount of data to a few monitorable quantities summarizing its quality and the status of the detection systems. Such quantities have been encapsulated in a unique interface called 'monitor element'. To evaluate the data quality, built-in template tests are provided and are frequently processed by the

¹Calorimeter for LLinear Collider Experiment

²Conseil European pour la Recherche Nuclaire

system. A user API is provided for the online analysis module to book *monitor elements* and their quality tests which are then published across the network. They are in their turn collected by server applications and distributed to visualization clients, again, either on manual query or in update mode.

To visualize monitor elements from the collectors, a graphical user interface has been developed using the Qt[1] toolkit. It can use multiple client interfaces to every available collectors on the network. The user can then display received *monitor elements* in areas organized in tabs. Multiple configurations with different sets of monitor elements can be set-up at once. This allows for a quick overview of all the critical elements and variables needed for the good operation of the experiment.

III. IMPLEMENTATION AND TESTS

In order to test the framework, a dedicated implementation has been developed for the CALICE collaboration and its SDHCAL and SiWecal prototypes. An interface to read and write events in the LCIO format has been written using a generic data streaming toolkit named *xdrstream*.

As mentioned previously, the CALICE-SDHCAL collaboration has developed tools to dump online raw data coming from the detectors to a shared memory space as data source buffers. To treat these buffers, a dedicated shm processor plugin has been implemented. It reads the data sent by the DAQ system into the shared memory, then call the online event builder (*levbdim*) to reconstruct data into the lcio format. The reconstructed event is finally sent over the network to the event collector.

The LCIO EDM contains multiple data structures for different levels of reconstruction that analysis can use. To overcome this problem, data converters are provided to pass from one LCIO data structure to the other. This gives the ability to plug off-line analysis into the monitoring system, leading to a better assessment of the overall data quality.

The monitoring system can also be tested with LCIO data files as a data source. In order to keep conditions as close as possible to a real setup, it can be configured to simulate the timing structure of the raw data.

For the two weeks beam test campaign at CERN, specific analysis aimed to the monitoring of the SDHCAL and SiWecal have also been developed. They come as independent modules to plug into the framework and can be started or stopped at any given time. This implies that even a buggy analysis that would come to crash will not interfere with the rest of the monitoring. All the modules are configurable either through xml files or command line: most of the parameters can be quickly adjusted, a few seconds prior to start the module.

Part of the monitoring includes a module to watch the slow control parameters (high and low voltages, pressure, temperature, etc.). As these information do not come from the DAQ system, it is developed as a standalone module (see right analysis module block in Fig. 1). One analysis is dedicated to the raw data study and permits the shifters to quickly discover and try to correct for noisy part of the detector. A beam study analysis is available to display metrics about the beam spill

such as its length, time structure, number of acquisition cycle per spill, etc. Some *monitor elements* from this module can also be used as performances indicators. One practical example would be the ratio between the number of events treated by the monitoring and the DAQ systems.

Other modules are dedicated to efficiency measurement of various parts of the detectors, particle identification or tracking. Specific information using core properties of the detectors are also recorded like the deposited energy (SiWecal) or number of hits for each threshold (SDHCAL). Finally event display modules are also present to visualise particle interactions inside the detectors.

The Marlin[5] framework being widely used within the Linear Collider Collaboration for data analysis, an interface is under development and will permit users to directly plug their analysis as it is into the monitoring.

The framework has been successfully tested with off-line SDHCAL data from previous beam test using the time structure of the beam spills. Successful online tests were also performed by taking cosmic muons data with the same deployment scheme foreseen for the full scale testing at CERN. For this campaign, the framework will be deployed across multiple computers, each one running a set of collectors (*monitor element* and event) and analysis modules. Several monitoring GUIs will be running on these computers and personal shifters equipment.

IV. CONCLUSION

In light of the critical importance of a tool to quickly detect problems and ensure a good data quality, a new generic framework for Data Quality Monitoring systems has been created.

Contrary to most systems available for High Energy Physics up to now, it is not tied to a given experiment set-up. All the tools needed to develop a specific implementation depending on specificities of the experiment, such as DAQ interface, data format and detector analysis, are provided within the framework.

To put to test this system, a dedicated implementation has been developed in parallel for the CALICE SDHCAL and SiWecal collaborations. The whole system has been successfully tested with the CALICE SDHCAL prototype with offline data and online cosmic muons. It will be put to full scale testing during a two weeks beam test campaign at CERN with the CALICE SDHCAL and SiWecal prototypes. The framework being in its final stage and given successful SDHCAL tests, satisfactory results are expected.

REFERENCES

- [1] Qt Company, <http://www.qt.io>, v4.7, 2016.
- [2] C. Gaspar et al., *DIM, International Conference on Computing in High Energy and Nuclear Physics* (Padova, Italy, 1-11 February 2000)
- [3] Michael J Hammel. *Mongoose: an embeddable web server in C*. Linux Journal, 2010(192):2, 2010.
- [4] I. Antcheva et al., *Comput. Phys. Commun.* **182**, 1384 (2011)
- [5] F. Gaede, *Marlin and LCCD: Software tools for the ILC*, *Nucl.Instrum.Meth.* A559 (2006) 177-180
- [6] Frank Gaede, lcio.desy.de, 2016.