Monitoring of the data processing and simulated production at CMS with a web-based service the Production Monitoring Platform (pMp)

Antanas Norkus for the CMS Collaboration

Abstract

Physics analysis at the Compact Muon Solenoid (CMS) requires both a vast production of simulated events and an extensive processing of the data collected by the experiment. Since the end of the LHC runI in 2012, CMS has produced over 20 Billion simulated events, from 75 thousand processing requests organised in one hundred different campaigns, which emulate different configurations of collision events, CMS detector and LHC running conditions. In the same time span, sixteen data processing campaigns have taken place to reconstruct different portions of the runI and runII data with ever improving algorithms and calibrations. The scale and complexity of the events simulation and processing and the requirement that multiple campaigns must proceed in parallel, demand that a comprehensive, frequently updated and easily accessible monitoring be made available to the CMS collaboration. Such monitoring must serve both the analysts, who want to know which and when datasets will become available, and the central teams in charge of submitting, prioritizing and running the requests across the distributed computing infrastructure of CMS. The Production Monitoring Platform (pMp) web-based service, has been developed in 2015 to address those needs. It aggregates information from multiple services used to define, organize and run the processing requests; pMp updates hourly a dedicated Elastic database, and provides multiple configurable views to assess the status of single datasets as well as entire production campaigns. This contribution will cover the pMp development, the evolution of its functionalities and one and half year of operational experience.

Presented at CHEP 2016 22nd International Conference on Computing in High Energy and Nuclear Physics
Monitoring of the data processing and simulated production at CMS with a web-based service: the Production Monitoring Platform (pMp)

G Franzoni\(^1\), A Norkus\(^{1,2}\), A A Pol\(^1\), N Srímanobhas\(^4\) and J Walker\(^3\)

\(^1\) CERN, PH CMG-CO; CH-1211 Geneva 23 Switzerland
\(^2\) University of Vilnius, Universiteto g. 3, Vilnius 01513, Lithuania
\(^3\) University of Sheffield, Western Bank, Sheffield S10 2TN, UK
\(^4\) Chulalongkorn University, 254 Phayathai Road, Pathumwan, Bangkok 10330, Thailand

E-mail: antanas.norkus@cern.ch

Abstract. Physics analysis at the Compact Muon Solenoid requires both the production of simulated events and processing of the data collected by the experiment. Since the end of the LHC Run-I in 2012, CMS has produced over 20 billion simulated events, from 75 thousand processing requests organised in one hundred different campaigns. These campaigns emulate different configurations of collision events, the detector, and LHC running conditions. In the same time span, sixteen data processing campaigns have taken place to reconstruct different portions of the Run-I and Run-II data with ever improving algorithms and calibrations. The scale and complexity of the events simulation and processing, and the requirement that multiple campaigns must proceed in parallel, demand that a comprehensive, frequently updated and easily accessible monitoring be made available. The monitoring must serve both the analysts, who want to know which and when datasets will become available, and the central production teams in charge of submitting, prioritizing, and running the requests across the distributed computing infrastructure. The Production Monitoring Platform (pMp) web-based service, has been developed in 2015 to address those needs. It aggregates information from multiple services used to define, organize, and run the processing requests. Information is updated hourly using a dedicated elastic database and the monitoring provides multiple configurable views to assess the status of single datasets as well as entire production campaigns. This contribution will describe the pMp development, the evolution of its functionalities, and one and half year of operational experience.

1. Introduction
The Large Hadron Collider (LHC) [1] at CERN is 26.7 km long two-ring superconducting hadron accelerator and collider. It is designed to collide proton beams with a centre-of-mass energy of 14 TeV and luminosity of \(10^{34}\) cm\(^{-2}\) s\(^{-1}\). The Compact Muon Solenoid (CMS) experiment [2] is an omni-purpose detector operating at LHC. The central feature of the CMS apparatus is a superconducting solenoid of 6 m internal diameter, providing a magnetic field of 3.8 T. Within the superconducting solenoid volume are a silicon pixel and strip tracker, a lead tungstate crystal electromagnetic calorimeter, and a brass/scintillator hadron calorimeter. Muons are measured in gas-ionization detectors embedded in the steel flux return yoke outside the solenoid. In addition, the CMS detector has extensive forward calorimetry. The analysis of LHC data in
CMS needs production of simulated collision events. During the runI of LHC (2010-2012), the CMS collaboration produced over 12 billion simulated events. Having an overview of the current state of production system is crucial in planning and managing it.

This paper describes the procedures and infrastructure used by CMS experiment to display and monitor the production of simulated events.

2. McM platform
The organization and production of simulated events are responsibility of the Monte Carlo Coordination Meeting (MCCM) team. A Monte Carlo Management (McM) platform [3] was developed to bookkeep and centralize the production of simulated samples. Currently McM contains information on:

- 127 campaigns highest level of abstraction where requests belonging to a campaign share same software version and major customisations;
- 407 flows defining how two campaigns are connected (e.g. one campaign is an input to second);
- 105967 requests user specific configuration where he defines physics process, needed number of events and customisation of parameters;
- 139905 workflows definition of request in CMS computing infrastructure, as single request can have multiple workflows due to resubmission or recovery.

The McM web service provided a number of plots to display overview some of the data. These plots were useful, but used the same system and database as McM service itself, and were of limited scope. The need to create a separate service which would not affect other services and would be accessible by anyone in CMS collaboration was clear.

3. pMp platform
The pMp web platform is developed to provide means of following the progress and status of the requests of events processing which are submitted to CMS computing infrastructure [4]. Its design and dependencies are shown in figure 1; it aggregates information from multiple services in the CMS computing infrastructure and visualizes data for user consumption. pMp uses its own database which is periodically updated from the input services, meaning that when a user loads information, his/her request does not interact with other services from CMS infrastructure.

While the pMp interface can be accessed by any member of the CMS collaboration, its main purpose is serving the needs of MCCM team: service administrators, generator contacts who collect the needs for simulated samples from within the detector of physics group they represent, request managers who check and approve the configuration of the processing requests, production managers who configure, check and submit requests to CMS production infrastructure.

pMp can display information for two different types of requests: requests of simulated events production, submitted via the McM platform, and data processing requests, which are submitted to CMS computing infrastructure via dedicated scripts.

4. Infrastructure and technical specifications
One of the most crucial need for the service is data consistency. This is achieved by storing data in an internal database and doing frequent incremental updates. This makes pMp not dependent on external services outages. To achieve this pMp use an elastic search database to index data from input services. Furthermore, all pMp plots needs to be flexible and easily configurable to provide specific information to each physics working group.
4.1. Databases
The pMp service uses two input databases: McM and Stats, both databases are implemented using CouchDB as a NoSQL solution. The stats database dynamically aggregates information from multiple sources for all requests submitted to CMS computing infrastructure. McM records information of all centrally produced simulated events. By using NoSQL which is a schemaless database, both services can save data without problems even when there is change or evolution of the input database schema.

The pMp service uses an elastic search database to collect and index the inputs from Stats and McM databases. We have implemented our own custom update utility named *fetchd* which uses a feature of CouchDB which allows to retrieve only the latest changes from a certain index point.

The need to have a system not dependent on other CMS services requires to run database updates as often as possible. This is achieved by running regular incremental updates indexing small amount of data each time by using previously introduced CouchDB feature and storing only the local index.

4.2. Back-end
The back-end is written using the Python programming language and the Flask web framework utilizing Tornado web framework to manage concurrent HTTP connections. The Flask framework is used to manage routing of incoming requests, and to implement a restful API. Thanks to the restful API, the pMp service can be accessed via web browser or with simple HTTP requests from any script. Data is returned as JSON objects which are human readable and easy to parse by machines.

We use Jenkins to automate tasks like database updates, integrity checks with notifications plug-in to send e-mail when a task fails to complete.

4.3. Front-end
To achieve customizable, bookmarkable, and simple to generate plots, we use frameworks that support the newest web technologies based on JavaScript, HTML5 and CSS. We used AngularJS to handle most of JavaScript tasks for data loading, updating, customizing or filtering data. To have the web pages look clean we used Bootstrap CSS. All pMp instances are running with Grunt task runner for automating tasks of concatenating, minimizing, uglifying Javascript code and Bower to manage all JavaScript and CSS dependencies.
5. Types of plots

The pMp service has 3 main plots and these will be explained in detail later. All the plots displayed in pMp web page needed to be easily and consistently exchanged across different CMS members and to serve the specific needs and interests of many detector and physics groups. Customisation can be various: from filtering the data to be shown to choosing how to stack information in bar charts.

(a) Present statistics.  
(b) Historical statistics.  
(c) Performance statistics.

**Figure 2.** Types of plots

5.1. Present statistics

The Present statistics bar chart in figure 2a; displays the current snapshot of a set of processing requests using histograms; The advancement of a request in the production system traverses different states; the number of events which are in status:

- done;
- submitted;
- defined and ready to be approved;
- of currently being validated;
- new events.

can all be showed selectively or inclusively. This plot has customizable grouping by specific parameter (status, priority, physics working group etc.). The Present Statistics page can display information for different level of abstraction: campaigns, chained requests, requests, workflows.
5.2. Historical statistics

The Historical statistics view present a time series as shown for example in figure 2b; shows events completion in production over time:

- the grey colored graph shows the time evolution of total number of events submitted for processing to CMS computing infrastructure;
- the orange graph shows the number of events already processed as part of requests which are not yet finished;
- the blue graph shows the number of events already processed as part of requests which have completed and whose datasets have been published for analyzers to use.

The plot is dynamically generated so it is easy to display data for user specific searches (e.g. campaign, request, or single workflow). On the same page, the service also displays the list of request and their completion percentage in production.

5.3. Performance statistics

The performance statistics are shown in figure 2c; this includes a bar chart, generated table and scatter plots intended to monitor the time needed for request to transit across its statuses, hence monitoring time devoted to the actual processing as well as the time needed by the operators involved in the whole submission process. The bar chart displays the distribution over time for all requests in a selected campaign. The table shows key statistics of campaign: time, range, total number of requests in campaign. The scatter plot presents the campaigns activity over time, there each dot represents single request and shows how it changed its status over time.

6. Summary

In this paper we have described the procedures and the infrastructure deployed by the CMS collaboration to submit and monitor processing request of acquired or simulated events. Two key web-based platforms McM and pMp are used to bookkeep, submit and monitor all steps of this process. We focused in illustrating monitoring part and its web-based platform, the technologies chosen to implement its features. The choice and design of NoSQL database to achieve fast incremental updates and data reliability was described. The different types of monitored quantities used by CMS community were explained in detail.

References
[1] Lyndon Evans and Philip Bryant, LHC Machine, 2008 JINST 3 S08001, doi:10.1088/1748-0221/3/08/S08001.
[2] CMS Collaboration, The CMS experiment at the CERN LHC, JINST 3 (2008) S08004, doi:10.1088/1748-0221/3/08/S08004.
[3] G Franzoni et al for the CMS collaboration Monte Carlo Production Management at CMS 2015 J.Phys.: Conf. Ser. 664 072018 doi:10.1088/1742-6596/664/7/072018
[4] E Fajardo et al for the CMS collaboration A new era for central processing and production in CMS 2012 J.Phys.: Conf. Ser. 396 042018 doi:10.1088/1742-6596/396/4/042018