Software and experience with managing workflows for the computing operation of the CMS experiment

Jean-Roch Vlimant, on behalf of the CMS Collaboration
California Institute of Technology
E-mail: jvlimant@caltech.edu

Abstract. We present a system deployed in the summer of 2015 for the automatic assignment of production and reprocessing workflows for simulation and detector data in the frame of the Computing Operation of the CMS experiment at the CERN LHC. Processing requests involves a number of steps in the daily operation, including transferring input datasets where relevant and monitoring them, assigning work to computing resources available on the CMS grid, and delivering the output to the Physics groups. Automation is critical above a certain number of requests to be handled, especially in the view of using more efficiently computing resources and reducing latency. An effort to automatize the necessary steps for production and reprocessing recently started and a new system to handle workflows has been developed. The state-machine system described consists in a set of modules whose key feature is the automatic placement of input datasets, balancing the load across multiple sites. By reducing the operation overhead, these agents enable the utilization of more than double the amount of resources with robust storage system. Additional functionality were added after months of successful operation to further balance the load on the computing system using remote read and additional resources. This system contributed to reducing the delivery time of datasets, a crucial aspect to the analysis of CMS data. We report on lessons learned from operation towards increased efficiency in using a largely heterogeneous distributed system of computing, storage and network elements.

1. Introduction
The Compact Muon Solenoid (CMS) experiment [1] is a multipurpose particle detector hosted at the Large Hadron Collider [2] (LHC) which delivers proton-proton collisions. The CMS detector consists of about a hundred million electronic channels clocked at 40 MHz. Signals from particles coming from the interaction regions are triggered and recorded at one kHz and processed in a real time pipeline. Collision data may be subsequently reprocessed when new calibration or software become available with improved overall physics performance. Analysis of such dataset requires a large volume of simulated collisions, in approximate ratio of 130% of collision events. The Monte-Carlo simulation (MC) are aggregated in several tens of thousands of datasets for a total of several billion of events. The design and operation of a component critical to the swift production of simulated events and the reprocessing of collision data is reported in this document. This sub-system was developed as an effort to consolidate CMS computing operation and is named Unified as it has regrouped several overlapping sets of computing operation procedures. This was deemed necessary to cope with the ever growing and diversified needs in production.

This paper is organized as follow. First, we provide an overview of the central production, then focus on the implementation and overall functioning of components. The strategies adopted
Figure 1. Diagram of the main components and actors of CMS central production. The system reported in this document is represented on the left box, interacting with all other components.

at several levels are then described, and we conclude with operation considerations and overall performance.

2. Central Production

2.1. Computing Infrastructure

The LHC Grid [3] is composed of more than a hundred computing sites of various size ranging from a thousand cores to couple of tens of thousands of cores. Computer centers are organized in tiers, which takes its definition in the earliest computing model of CMS. The Tier0 (T0) is primarily focused on the real time processing of the detector data [4] and used opportunistically for central production. There are 7 Tier1 (T1) which provide tape storage (so is T0) in addition to compute power. There are about 60 Tier2 (T2) sites that are only compute and storage. While the early CMS computing model envisioned the use of the tier tree, it has become more like a full mesh cloud model over the world wide research network with a total of about 200,000 cores shared between central production and analysis. Large amount of opportunistic resources are being included under specific and dedicated sites [5]. This cloud of computer centers is by construction highly heterogeneous in the hardware available, network capacity, storage space and performance making the task of optimizing its usage a daunting one. We present in this paper a strategy and automation that tend to provide good usage of resource and fulfill the research goals of CMS.

2.2. Production Components

As shown in figure 1, there are four main contributors to the preparation of production.

- The generator group is in charge of the software and configuration specific to simulating the physics processes (only relevant for simulation).
- The alignment and condition group provides calibration constants required for data and simulation.
- The software group provides the simulation and reconstruction software (see section 2.3).
- The computing group provisions the resources necessary to meet the goals.
Three main services are used to organize the production, as can be seen in figure 1.

- All the ingredients for production are entered in the Monte-Calor Management system (McM) [6] in the form of requests, chains of request, campaigns and chains of campaigns. A request consists of the configuration how to run the CMS software, computing requirement parameters and book keeping information. A campaign is a set of requests with similar configuration made for a specific goal.

- Workflows containing one or more request are injected in request manager [7]. It produces workloads that have to be assigned to sites for processing. All the job splitting, job submission, retries, data book keeping and publication are handled in request manager and the production agent, described in details in [7].

- Jobs are submitted to HTcondor [8] which runs jobs at sites under the glide-in-wms scheme [9], where pilot jobs are submitted to run on local site batch queues and subsequently run jobs from a global pool. HTCondor is handling the matching of job requirement to site capabilities. HTcondor provides partition-able [10] computing slots with most generally 8 cores and 16GB of RAM available. These slots are partitioned based on job requirements.

The system detailed in this paper (see figure 1) is driving the data location, workflow assignment and job routing by bringing together McM, request manager and HTcondor. The goals are to minimize operation, maximize throughput and respect the priorities set by the physics coordination of the experiment to meet analysis goals.

2.3. Production workflow

The simulation of collision events are usually split in five stages that involve different software and requirements.

- Event generation (GEN, MC only) : involves external generator software [11] with dedicated interface to the CMS framework. This processing is dominantly very fast and requires no input data. This CPU-bound processing would be very suitable for HPC-like computer centers [12].

- Detector simulation (SIM, MC only) : involves running GEANT4 [14] software. This step canonically takes about a minute per event to simulate the trajectories of particle through the CMS detector. This step is commonly coupled to the GEN step. The input data that is occasionally needed is very small and is not a computing challenge.

- Signal digitization (DIGI, MC only) : is performed using dedicated CMS software which simulates the electronic response, including detector noise. It includes also the simulation of multiple interactions per bunch crossing happening in the LHC called pileup (PU). This latter part involves secondary input data and is performed using two methods. A legacy method is reading as many secondary events as required to simulate the overlay per bunch crossing, including out-of-time bunches. With an average PU of 40 and 12 bunch crossing to be considered, this amounts to more than 400:1 events overlay per primary event, resulting in an heavy read on the secondary input. The more recent method [15], developed to cope with ever increasing amount of pileup in the LHC, consists in running the legacy method only once per campaign to produce a large data bank of already mixed events. This event library is stored in a lightweight format and results in a much lighter read requirement since it requires only a 1:1 mixing. The challenge comes with the size (in the range 0.5-1PB) and accessibility of this secondary input, which is commonly read from remote storage through the network using xrootd [16] in the AAA federation [17].

- Event reconstruction (RECO) : consists of physics driven software [18] that extract the physics content from the detector data. This step takes of the order 15 seconds per event
or more depending on the LHC condition and the type of event. This stage is commonly coupled to the DIGI step for MC and is not read-intensive for data.

- **Analysis format encoder (AOD)**: is a rather fast step which provides in output a data format that contains the condensed information required for most analysis [19]. The input data is usually light, but because the software is fast, it can make the read requirement high.

These stages have to be separated chronologically when production has to start for scheduling reason and the software for later stages is not yet ready. Whenever possible, all stages are combined in a unique computing workflow so as to limit the amount of manipulations. Because of the very rich scientific program of CMS, many different variations of each stages have to be produced simultaneously resulting in an heterogeneous work load.

The workflows submitted can be categorized in three classes with regards to the input that they require.

- Workflows that read a privately produced set of LHE files [20], for which the generator has yet to be integrated to the CMS software.
  Usage of LHE file is not recommended and is not the main stream of production. These privately produced LHE files are stored at CERN and cannot be exported. Therefore, workflows of that kind are ran at CERN. Having only one production site for this kind of workflow is not optimum and can lead to long processing queues; the amount of resource reachable for subsequent jobs are extended to other sites reading over the network [16].

- Workflows that need secondary input data (PU simulation).
  Sites with robust storage, which can sustain high read on secondary, are included in a pool of sites that can participate in this activity. Future development in monitoring site performance and especially on network, local read and remote read would allow to dynamically categorized sites and fine tune the workload-to-site matching.

- Workflows that do not need any input data (typically event generation workflow) or no secondary.
  All possible production site are considered as a possible host for jobs.

3. Implementation

The state machine (see diagram in figure 2) reported in this document is composed of multiple modules performing a task specific to a given status. Internal status and various bookkeeping information are kept in an Oracle database hosted by CERN IT [21]. The database allows for multiple concurrent state transition on separate objects. No locking is implemented for concurrent update to individual entry and is prevented by running serially the modules acting on the same initial state. A multitude of transient and non-vital information is stored on disk in the form of json files produced for book-keeping, monitoring and driving purpose. Monitoring information are exposed using a standard http server. Logging about components and workflows are collected in an elastic search [22] instance and further exposed in an organized fashion under a standard http server.

The statuses shown in figure 2 have the following meaning:

- **considered**: defines when a workflow has been submitted and ready to be handled.
- **staging**: shows that either the primary or secondary input is being transferred to the production sites.
- **staged**: represents the fact that all the input requirements are met and the workflow is ready to be started.
- **away**: is set when jobs are being produced and handled within the production agents and HTcondor. The progress of jobs is monitored for dynamic intervention.
Figure 2. Diagram of the state machine for driving workflows through the computing infrastructure. The boxes represent the statuses and the arrows the possible transitions and specific actions.

- **assistance**: categorize workflows that have finished and have issues preventing from moving further. Issues include not having a transfer to tape, missing statistics, book keeping inconsistencies ...
- **close**: indicates that all the enforced data quality requirements are met and the outputs can be announced.
- **done**: labels that all data was announced and further archiving of production statuses is initiated.
- **forget**: tags workflows that are not of any relevance anymore.
- **trouble**: represents workload that is matching logical issue in production and needs replacement or further action.

4. Mode of Operation

4.1. Data Placement Strategy
Optimizing the LHC grid resource usage for data-intensive processing is a challenging task, and we compromised in using a strategy which should allow good usage, with adjustments performed dynamically. The base strategy for data placement consists of distributing the input data to as many sites as possible, following the amount of CPU available for central production. Large input datasets are sliced and each part is considered separately for distribution. This scheme assumes an already evenly distributed load of work over sites and loosely enforces that no too large amount of work would be send to a single site, therefore avoiding bottleneck in processing and delays in delivering the output. Backlog and delays can occur in case of issues with data transfer between sites, data corruption, network outage, storage downtime, ... Such backlog could be prevented using network and storage monitoring and is subject to future improvement. Depending on the processing strategy decided, it might be worth having more than one copy
of the primary input dataset spread over many sites. This does allow to have more resource available for a specific workflow, but requires more time for data transfer.

The secondary input location is handled upon configuration to either be at all candidate production sites or only at a selected few as source for remote read.

The monitoring and management of storage space for central production is fully delegated to the CMS dynamic data management system [23], and a locking mechanism is in place to prevent spurious deletion of the required data. Only a fraction of the provided quota is used so as to leave sufficient buffer for the output data, which in addition to be available in local storage is systematically consolidated in a full copy at one of the site contributing to the processing.

4.2. Data Integrity Diagnostic
Monitoring of the evolution of transfers is performed and issues are automatically reported. Data integrity is performed upon checking on availability and transfer of data. Inconsistency are reported for investigation. Further automation of this procedures is subject of future work, for example considering production acceptance (see section 4.6) to discard unusable data (corrupted, unreachable, lost, ...).

4.3. Workload assignment
Once the optimum requirements on the input datasets are met, the workflow can be assigned to production sites. The condition is loosened in case some of the transfers are problematic and there is already at least one copy of the primary input reachable on disk, or at least enough data to meet the closing criteria (see further in section 4.6 for details). The requirements in terms of number of threads and memory are used to filter sites. Only sites that are holding full or part of the input are used as target for jobs. Sites might further be added dynamically as described in the next section.

4.4. Resource Usage Optimization
Remote read is not systematically used on primary input dataset (while it may for secondary input). We resort to remote read if a backlog of work accumulates at a given site, or to expedite higher priority work. In both cases, we use remote reads to expand resources available to individual workflows. We define a set of source-destination pair for remote read using experience on network capacity. Future work on monitoring will allow to create this mapping dynamically. The workflow requirements are further used to filter the selection of candidate destination sites. Technically, we use the HTCondor job router [8] to apply these rules to pending jobs. The job routing is further used to extend processing to resource dynamically made available.

4.5. Workload Requirement
Jobs are submitted with estimated requirements in terms of memory and runtime. Upon completion of a sufficient number of jobs within a subcategory, the 95 percentile of memory and walltime are measured and use dynamically to edit the remainder of jobs. Memory is always set to lower values so as to optimize the partitioning of multicore pilots. The wall time requirement is always set to a higher value, to prevent the risk of jobs exceeding the lifetime of the pilots they run in.

4.6. Completion Strategy
The system allows for flexibility in setting up rules for declaring output data ready for announcement. Several requirements are fully enforced to prevent data corruption, and any deviation from the standards are reported. Further automation of operation should not be done because corruptions are to be understood and are symptoms of bad components in the system.
The valid output datasets are both distributed to the analysis disk pool and a copy is made to tape storage, when relevant. The truncation of the processing is automatized with configurable requirements.

4.7. Recovery of Failures
Despite all care taken to maintain high quality at all sites by administrators, distributed computing comes with some level of inefficiency. Most failures are handled in retries within the request manager or HTcondor. However, for those that are not handled automatically (storage failures, configuration error, transient network issue, ...), operators need to take action. To this end, detailed and focused error reports are generated to facilitate the work of operators and leave time for investigation of issues with experts. As much logs as possible are provided over standard http.

The system can be configured to automatized some of the understood errors such as exceeding memory. Automation with use of machine learning is subject to future work.

5. Operation and Performance
This system was deployed late 2015 and further developed to match the evolution in production strategy (such as using premixing for pileup simulation). Production campaigns need to be configured and monitored closely at the beginning and can then be set in automated mode. Over the couple of years of operation, an average of 2000 datasets were produced weekly, with little intervention. A weekly peak at about 5000 analysis datasets was reached during MC reprocessing prior to conferences. This represents about 3 billion of events produced per month averaging over all types of workflows. The system presented here has significantly reduced the amount of operation effort. Most of the production goes without operator’s intervention, leaving more time for investigation of failures and consolidations.

6. Discussion
The framework described in this document was developed primarily to scale the computing operation up to the ever growing complexity and volume of production at CMS. It is an incubator for ways of distributing work over to sites with a view to further integration in the production infrastructure. The build-in flexibility allows to accommodate for changes of production strategy and requirements. Automation increases the overall throughput, has relieved operator from tedious work and allows to concentrate on the improvement of the overall performance. Development is on-going to include prediction techniques based on machine learning algorithms. Future work may include mechanism to reduce the footprint on disk storage which will require more granular management of I/O data.

The vision is that a more integrated compute-network-storage elements system would allow for fine optimization and better overall usage and throughput with the allocated resources.

References
[1] Chatrchyan S et al. CMS Collaboration 2008 The CMS experiment at the CERN LHC J. Inst. 3 S08004
[2] Evans L and Bryant P 2008 LHC Machine J. Inst. 3 S08001
[3] Knobloch J et al. 2005 LHC Computing Grid Technical Design Report CERN-LHCC-2005-024
[4] Hufnagel D - The CMS TierO goes Cloud and Grid for LHC Run 2 - J. Phys.: Conf. Ser. 664 (2015) 032014
[5] S Fues et al. - The HEP Cloud Facility: elastic computing for High Energy Physics. This Conference
[6] Boudoul G and Franzoni G and Norkus A and Pol A and Srimanobhas P and Vlimant J-R - Monte Carlo Production Management at CMS - Journal of Physics: Conference Series Conference Series 2015
[7] Wakefield S et al. 2012 J. Phys.: Conf. Series 396 032113
[8] Thain D, Tannenbaum T, and Livny M, "Distributed Computing in Practice: The Condor Experience" Concurrency and Computation: Practice and Experience, Vol. 17, No. 2-4, pages 323-356, February-April, 2005
[9] Letts J et al. 2015 J. Phys.: Conf. Series 664 062031
[10] Perez-Calero Yzquierdo A et al. 2016 - CMS readiness for multi-core workload scheduling. This Conference
[11] http://indico.cern.ch/event/454993/
[12] Bendavid J et al. - Large-scale distributed usage of NLO and multi-leg Monte Carlo event generators on the grid and at Argonne Leadership Computing Facility. This Conference
[13] Ivantchenko V et al. 2016 - CMS Full Simulation Status. This Conference
[14] Banerjee S et al. 2016 - Validation of Physics Models of Geant4 using data from CMS experiment. This Conference
[15] Hildreth M A New Pileup Mixing Framework for CMS Proceedings of this conference CHEP2015
[16] http://xrootd.slac.stanford.edu
[17] Bloom K (for the CMS Collaboration) 2014 J. Phys.: Conf. Series 513 042005
[18] Lange D et al. - The CMS Reconstruction Software Journal of Physics: Conference Series 331 (2011) 032020
[19] Petrucciani G, Rizzi A, Vuosalo C - Mini-AOD: A New Analysis Data Format for CMS J.Phys.Conf.Ser. 664 (2015) no.7, 072052
[20] Alwall J 2010 A standard format for Les Houches Event Files Comput.Phys.Commun 176 300-304 2007
[21] https://www.oracle.com/database/index.html
[22] https://www.elastic.co/products/elasticsearch
[23] Iiyama Y et al. 2016 Dynamo - The dynamic data management system for the distributed CMS computing system. This Conference