CMS MONTE CARLO PRODUCTION IN LCG

J. Caballero, J.M. Hernández, P. García-Abia, CIEMAT, Madrid, Spain

Abstract

We have introduced novel concepts in the Monte Carlo production system of CMS which have made running the full production chain possible on LCG, from the generation of events to the publication of data for analysis, through all the intermediate steps. We have also coupled production and the CMS data transfer system and made the tools more robust, significantly improving the performance of production in LCG.

INTRODUCTION

Monte Carlo production is crucial for delivering large samples of fully simulated events required for detector performance studies and physics analysis. Until recently, production was done at large computer farms hosted by computing centers at CMS institutions.

CMS software experts have been trying for some time to port the production tool McRunjob [1] to the LCG [2], in order to utilize the large amount of computing, storage and network resources made available by LCG. While LCG provides the basic services for distributed computing, reliability and stability still remain the main problems, making a robust production system necessary.

An earlier implementation of McRunjob for LCG succeeded in running simulation jobs on LCG-1 [3], with a somewhat low efficiency. However, the digitization and reconstruction steps resisted these efforts.

This note describes the novel concepts that we introduced into McRunjob for LCG, which made it possible to improve the efficiency of simulation jobs and to run digitization and reconstruction jobs in LCG. Technical details on implementation and usage are found in the McRunjob-LCG documentation [6].

CMS production has been running in LCG for few months. The analysis of the performance of this production is hereby disused.

ATOMIC METADATA ATTACHMENT

The basic steps in the CMS production chain are: generation, simulation, digitization and reconstruction. The output of the simulation, digitization and reconstruction jobs is a set of three files (two for reconstruction) per job, called *EVD* files, each containing different portions of the events.

In local farm production, at the completion of the simulation step of a full dataset, the metadata attachment operation reads the EVD0 files (locally accessible via POSIX I/O) in order to generate the COBRA metadata, which are then stored in the *virgin META files*. The resulting *attached META files* are needed by the digitization and reconstruction jobs in order to process the simulated (and digitized for reconstruction) data.

In LCG, the output of the jobs is distributed among several remote *Storage Elements* (SE) with no POSIX I/O access. The overhead for collecting the EVDs at a single site, prior to the metadata attachment, is impractical given the amount and size of the files.

The *atomic metadata attachment* is simple and elegant: the metadata attachment is performed at runtime on the *Worker Node* (WN) only for the single run to be processed by the job. This operation introduces a negligible overhead, as the input EVD files are downloaded from the SE at startup. For reconstruction jobs, both the simulation and digitization metadata have to be attached.

In order to analyze the data, attached metadata have to be produced for the whole collection of simulated, digitized and reconstructed events. The EVD files of the different production steps are harvested by the transfer system and transferred to the analysis sites, where the metadata attachment is performed.

OUTPUT ZIP ARCHIVES AND PUBLICATION OF DATA

In the early attempts to run production on LCG, we found that one of the reasons for the low efficiency was the failure of the input/output (I/O) operations, with either the SE or RLS not being accessible. The number of ‘copy’ operations required (one per EVD file) increased the risk of failure.

We decided to pack the output EVD files into an uncompressed ZIP archive. As a consequence, the number of files to be copied to the SE is divided by a factor of three, reducing the complexity of the file transfer and management in production. Furthermore, the file size is significantly increased, solving the long standing problem of having too many small files for the file transfer and mass storage systems.

We further reduced the risk of losing the output due to I/O problems by using a user-defined list of backup SEs to which the output could be copied in case of failure of the copy to the reference SE. The temporary unavailability of RLS would make the copy operation fail for all the SEs, so this process was placed in a loop with a waiting time between iterations. The latency introduced this way may, in some cases, allowed RLS to recover.

The implementation of the ZIP schema had implications in the job preparation for the next production steps, as the production system was originally designed to deal with individual EVDs (not zips). We instrumented McRunjob to deal with zip archives properly.

One important consequence was the need to modify the CMS data publication tool (CMSGLIDE [8]) to be able to create POOL [7] XML catalogs and attached metadata files for production zipped archives. The publication of the data performs the global metadata attachment of the full dataset using zips directly without downloading/unzipping them.

TREATMENT OF PILE-UP

Proper simulation of events, reflecting the experimental conditions of LHC, requires the superposition of events (from inelastic pp interactions) on the events of the simulated physics processes. Technically this implies the preparation of pile-up samples with a large number of events, with typical sizes of the order of 100 GByte.

The pile-up sample (EVD and META files) is prepared at CERN and transferred to the T1/T2 centers that will run digitization jobs. A local POOL XML catalog for the pile-up, containing the *physical file names* (PFNs) of the local storage, must be created by the system administrators of the site and placed in a standard location. In addition, LCG sites with the pile-up installed this way, publish a *software tag* in the Grid information system which is used as a requirement of the digitization job. We have instrumented the job wrapper to search for the pile-up catalog at that location and merge it with the POOL XML catalog of the job.

COUPLING WITH THE DATA TRANSFER SYSTEM

Production EVD files stored on LCG SEs must be made known to the CMS data transfer and placement system, PhEDEx [5], in order that they be reliably and efficiently transferred. In the process of *data injection* into PhEDEx, file attributes are inserted in the PhEDEx central transfer management database (TMDB). PFNs are only available in the local PhEDEx catalog running at the sites hosting data. In the LCG case, this catalog is the LCG central catalog (either RLS or LFC).

A *virtual* LCG PhEDEx node comprises all LCG sites storing production data. Production files are injected at this node as they become available. The LCG PhEDEx node allows the harvesting of production files so that they can be transferred to other real PhEDEx nodes. PhEDEx Routing and Export agents for this node run somewhere centrally. Several instances of the LCG injection agents can be run at different sites, typically one at every UI machine submitting production jobs to LCG.

Files are *atomically* injected run-by-run (on a per job basis) as they become available from production. This way, files are available for transfer as they are produced. There is no need to wait until the whole collection has been processed and published as available.

The link between production and PhEDEx is done through the production *summary file* [1], which contains all the information needed to inject the produced data into PhEDEx, namely, the file POOL attributes (GUID, *logical file name* or LFN, etc.), checksum and filesize. The injection agents extract that information and store it in the PhEDEx database. The summary file of a production job is stored in the job output sandbox, which must be retrieved by the production manager so that the summary file can be processed by the injection agent. The zip archive containing the EVD data files is the one injected into PhEDEx.

OTHER FEATURES

Reduction of the input sandbox size

At job submission time, the input sandbox is sent to the *Resource Broker* (RB), from where the jobs are dispatched to *Computer Elements* (CE). In early implementations of McRunjob for LCG, all the auxiliary files required for the job to run were sent together with the jobs in the *input sandbox*. This included setup scripts and the COBRA virgin META files.

Large input sandboxes introduce a non negligible overhead in the job submission and cause problems at the RBs. To prevent the disk space and network bandwidth of the RB becoming saturated by massive job submissions (as expected in large scale productions), there is a limit on the size of the input sandbox of 10 MByte. This limit is just right for simulation jobs ($\sim$9 MB), but too low for digitization jobs (up to 20 MByte). We decided to take all the large files out of the input sandbox and place them on the SE (operation performed only once per production step and dataset), reducing the size of the input sandboxes to about 20 KByte. This reduced significantly the time necessary for submitting the jobs.

Local software installation

The software of the experiment is installed on the LCG CEs by the software manager, in a shared filesystem (normally NFS) accessible by all WNs. In some cases, this software might be wrongly installed or temporarily inaccessible, causing the jobs to crash.

We have made an experimental version of McRunjob that verifies the availability of the software at runtime. The software required is downloaded from a SE [2] and installed locally. This local software installation at runtime is verified before the application starts.

We have used this mechanism to run official production at sites without the CMS software installed, with promising results. Software installation at runtime would also allow

[1] This file contains validation information of the job, which is stored in the central production Database at CERN (RefDB).

[2] The production manager stores replicas of the CMS software in several SEs and registers then in RLS.

production jobs to be run at sites that do not provide specific support to CMS but make their CPU resources available to the collaboration.

Local Pile-Up installation

As explained previously, the pile-up requires a specific installation at the sites destined to run digitization. This reduces the chances of exploiting Grid resources which are not under the control of the experiment: new sites, sites with little or no local support, etc. In sites with the pile-up installed, the performance of digitization jobs is limited by the maximum number of jobs that can concurrently access the pile-up from the storage system.

We have implemented a solution based on the ATLAS operation mode. The pile-up sample is stored (and replicated) in several SEs. The job wrapper selects a random subset of pile-up runs and downloads the corresponding files. The pile-up metadata attachment is done only for those pile-up runs which have been selected. The number of runs has to be defined to compromise between the size of the sample to be transferred through the network and the required minimum number of pile-up events to maintain fidelity of the simulated physics.

This implementation has been successfully tested, but has not yet been applied to physics studies.

PRODUCTION OPERATIONS

Production operations in LCG started about a year ago, In the first phase, it was driven from CIEMAT by about 1.5 FTE, other production operators joining the effort few months later. The number of events (in millions) produced in LCG per data tier are: 13.1 generated, 11.7 simulated, 11.4 digitized and 5.1 reconstructed. The accumulated number of simulated and digitized events is shown in Fig. 1 as function of time.

The efficiency of production depends significantly on the stability and reliability of LCG and the sites. Production was run using white lists, in which large and robust sites are included, as well as those with pile-up available. Production jobs run at different sites, as shown is Fig. 2 for about 30% of the production in LCG. Large sites like CNAF, RAL and DESY, joined the production of digitization and reconstruction jobs only recently. The full power of those sites is currently being exploited for the new production requests.

Despite the improvements made to the production framework, the efficiency resulted lower than the expectations, about 70%. The main reasons of failure were related to the stage in (25%) and out (16%) of files, temporary unavailability of RLS (6%), problems accessing the pile-up (19%) and other problems related to LCG and the sites (25%). The number of times a job had to be submitted before it succeeded is displayed in Fig. 3 (top), for a fraction of the production.

The weakest points of the CMS production system in LCG have been identified as the lack of an automatic monitoring and resubmission system, the lack of coupling of production and the CMS data management system (no possibility of pre-staging input files) and the lack of manpower. The temporary grid and site problems (CE, SE and RLS related) had an important impact in the efficiency, together with the lack of dedicated resources and priorities, as production jobs had to compete with CMS analysis and other experiments' jobs.

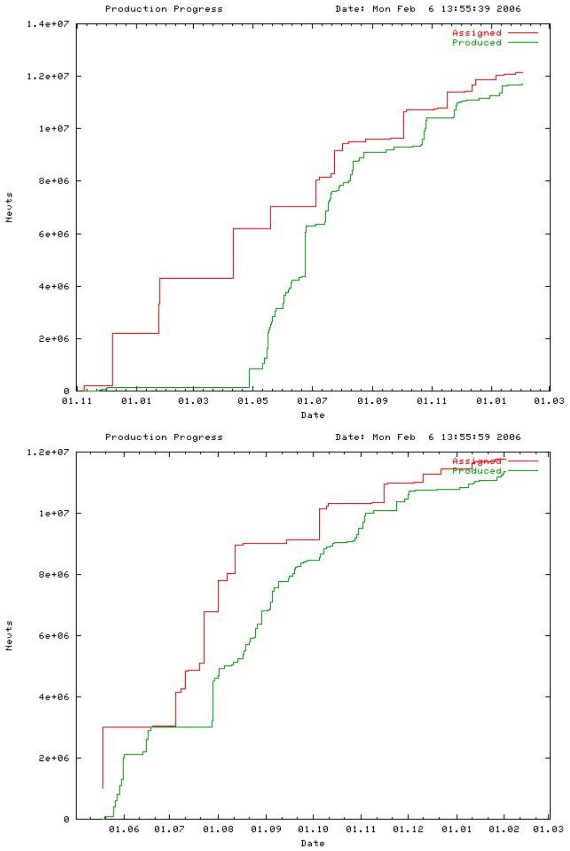
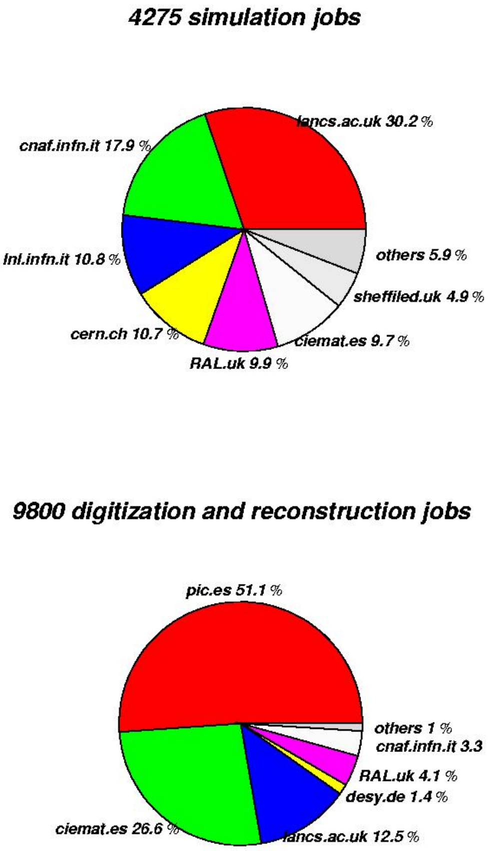


Figure 1: Number of (top) simulated and (bottom) digitized events, as function of time, accumulated until Feb.'06. The top graph starts on Nov.'04 and the bottom one on June '05.

Recently, CMS has migrated from RLS to LFC as a global file catalog for LCG. We adapted McRunjob to use LFC instead of RLS. So far, only a small fraction of the production in LCG has been done using LFC, however the results indicate a significant improvement in performance (90%) as compared to RLS, as observed in Fig. 3 (bottom).

CMS is currently developing a new Monte Carlo production system which incorporates the novel ideas and concepts we have introduced in the production framework, highly benefiting from our experience running production in LCG. The design of the new production system includes automatic data merging step, job chaining and coupling with the Data Management System. The new Event Data Model of CMS eliminates important constraints on modularity (like the metadata attachment) and has a more robust error handling system, which allows for an improved mon-


4275 simulation jobs

9800 digitization and reconstruction jobs

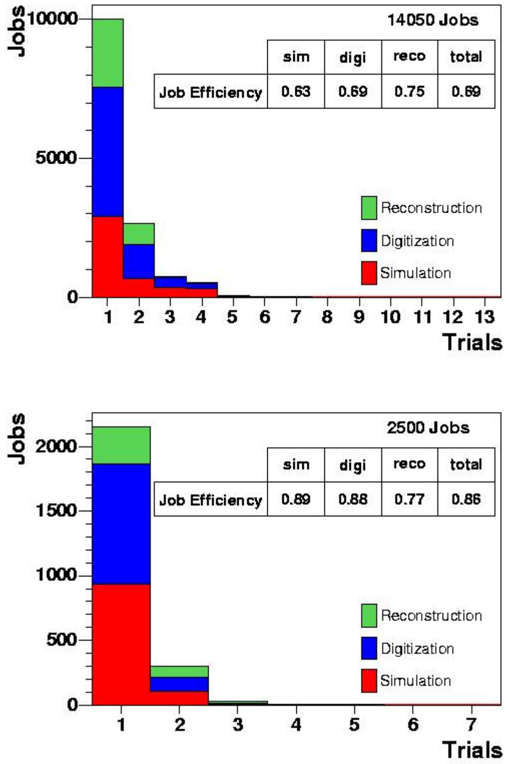
Figure 2: Contribution of different sites to the production of simulation (top), digitization and reconstruction (bottom) jobs. The figures correspond to about 30% of the production in LCG.

itoring and a higher degree of automation of production.

CONCLUSIONS

We have successfully made an "end-to-end" implementation of the CMS Monte Carlo production system on LCG-2, from the generation of events to the publication of data for analysis, through all the intermediate steps. The introduction of atomic operations like atomic metadata attachment and injection in the transfer system has been the key to our success. We have made several operations more robust, so improving significantly the efficiency. Novel developments for making jobs independent of the environment (local software and pile-up installation) will also help to further improve efficiency and make LCG a more reliable environment.

We have used the McRunjob tool to produce several million events on LCG-2 for different production steps and physics channels. This was the first time CMS digitization and reconstruction jobs have run on LCG. We have made a complete performance analysis, extracting important conclusions that will be the guidelines of the new Monte Carlo production system, based on MCPS [9].



	sim	digi	reco	total
Job Efficiency	0.63	0.69	0.75	0.69

	sim	digi	reco	total
Job Efficiency	0.89	0.88	0.77	0.86

Figure 3: Distribution of the number of times jobs had to be submitted to the grid before they succeeded, for the production performed with the RLS (top) and LFC (bottom) global file catalog. The job efficiency is also indicated for each data tier in each case.

REFERENCES

[1] Runjob Project, *http://projects.fnal.gov/runjob/*.

[2] LHC Computing Grid, *http://cern.ch/LCG/*.

[3] P. Capiluppi et al., CMS NOTE-2004/034, "CMS Results of Grid-Related Activities Using the Early Deployed LCG Implementations".

[4] COBRA, "Coherent Object-oriented Base for Reconstruction, Analysis and simulation", *http://cern.ch/cobra/*.

[5] PhEDEx, "Physics Experimental Data Export", *http://cern.ch/cms-project-phedex/*.

[6] P. Garcia-Abia and J.M. Hernández, "Running CMS Production on LCG", *http://wwwae.ciemat.es/cmsprod/McRunjob/doc/*.

[7] POOL, "Pool Of persistent Objects for LHC", *http://pool.cern.ch/* and *http://pool.cern.ch/talksandpubl.html*.

[8] M.A. Afaq, Publish Service CMSGLIDE, *http://home.fnal.gov/∼anzar/CMSGLIDE/* and *http://lynx.fnal.gov/runjob/Setup_20Publish_20Service*.

[9] The MCPS Project, *http://www.uscms.org/SoftwareComputing/Grid/MCPS/*.