PRODUCTION MANAGEMENT SOFTWARE FOR THE CMS DATA CHALLENGE

J. Andreeva, *CERN, Geneva, Switzerland*

N. Darmenov, V. Lefebure, *CERN, Geneva, Switzerland*
P. Garcia-Abia, *CIEMAT, Madrid, Spain*
A. Anzar, G. Graham, G. Guglielmo, N. Ratnikova, *FNAL, Batavia, USA*
A. Fanfani, C. Grandi, *INFN-Bologna, Italy*
T. Wildish, *Princeton University, USA*

Abstract

This paper describes CMS production and management software components, their functionality, implementation, communication between them. The main focus is given to the use of the CMS production system during data challenge of 2004 (DC04).

CMS DC04 TASKS AND ROLE OF THE PRODUCTION SYSTEM

The aim of the CMS data challenge of 2004 was to achieve following goals :

- Simulate a sustained data-taking rate equivalent to 25 Hz at a luminosity of 2x10^34 cm $^{-2}$ s $^{-1}$ for one month, which would correspond to 25% of LHC startup.
- Transfer data files produced at the Tier-0 to the Tier-1s
- Run analysis tasks on the data files arrived at Tier-1 as soon as data becomes available there, so called "real time" analysis [1]
- Run user analysis at Tiers-1s and Tier-2s as soon as data collections get published to the CMS physics community

The production system was responsible for the first task, which in its turn represented two major steps :

- pre-challenge production to provide input data for the reconstruction
- reconstruction aimed to simulate data-taking and to provide data for the physics analysis.

PRE-CHALLENGE PRODUCTION

The CMS DC04 pre-challenge production started in July 2003. It included generation, simulation and digitization production steps. It was run in a distributed heterogeneous environment. More than 35 CMS regional centres in Asia, Europe and USA took part in the pre-challenge production. Jobs had been submitted to a variety of the local schedulers as well as to two different flavours of Grid – Grid3[2] and LCG2 [2][3].

One of the important tasks of the pre-challenge production was to transfer output files from regional centres to CERN (Tier-0) to provide the input for the reconstruction step which was run at the Tier-0.

Input preparation for the reconstruction did not only include checking of data integrity and recording input files on CERN Mass Storage System (Castor) [4], but also some CMS specific procedure - so called "publishing".

While reading data collection, in addition to availability of the data files, CMS reconstruction framework (COBRA) [5] requires a set of META files containing references to the internal objects recorded in the data files and a POOL [6] catalogue with logical file names, physical file names and meta attributes of data and meta files belonging to a given collection. Creating of collection- related COBRA meta files and of POOL XML catalogue is known as the "publishing" procedure.

The period 2003-2004 was a time of intensive development of the core part of the CMS software. CMS dropped Objectivity and passed to a new persistency system, POOL. The fortran-based simulation program was replaced by C++ simulation (OSCAR) [7] based on Geant4. Delays in the entire software chain from POOL to the CMS reconstruction application caused the delay of the pre-challenge production and data challenge start-up. According to the original planning, 50 million events requested by CMS physicists had to be generated, simulated and digitized by November 2003, data files had to be shifted to Tier-0 in November-December 2003 with the rate of 1 TB per day for 2 months (non-trivial at the time), all input data collections had to be prepared (published) for reconstruction. The data challenge should have started in February of 2004. In reality more than 75 million events were requested by CMS physics community. These requests represented real data requests not just for the data challenge but also to be used for the analysis for physics TDR. 50 million events had been simulated by the beginning of 2004 . It was possible to prepare only about 10 million digitized events before the start of DC04, which began on the 1st of March 2004 without any rehearsal . Digitization was run in the background during the data challenge at a rather high rate (See Figure 1).

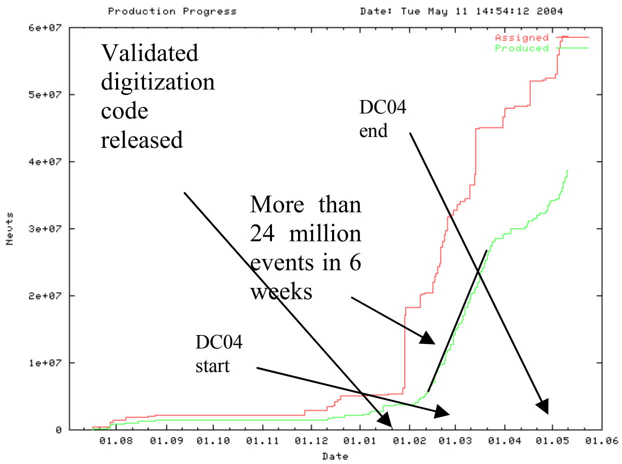

Figure 1: Production progress for digitization requests July 2003- May 2004

The large scale of the requested/produced data as well as the heterogeneous environment both contributed to the complexity of the production software system.

SUBSYSTEMS OF CMS PRODUCTION PROJECT (OCTOPUS)

The CMS production software project is called OCTOPUS.

It consists of the following components :

- **RefDB** [8] – the CMS Monte Carlo Reference Database
  Consists of MySQL database and WEB interface to it , written in PHP and CGI.
- **McRunjob** [9] – Monte Carlo Run Job, workflow planner for production processing.
  It is an OOPython framework for creating/submitting of large batches of production jobs
- **BOSS** [10] - Batch Object Submission System
  A C++ program for local book-keeping and real –time monitoring, uses MySQL for recording of the monitoring information.
- **UpdateRefDB** – perl module running at CERN as a crontab job and updating RefDB with the meta information sent by every successful job
- **DAR** [11] – Distribution After Release
  Distribution system for CMS application software, provides a tool supporting distribution of a required version of a given CMS physics application to the production regional centers. Used both for creation of the distribution and its installation.

The responsible for the physics group submits a request using web interface to RefDB, Production manager assigns a request to a given regional centre taking into consideration data and resources availability.

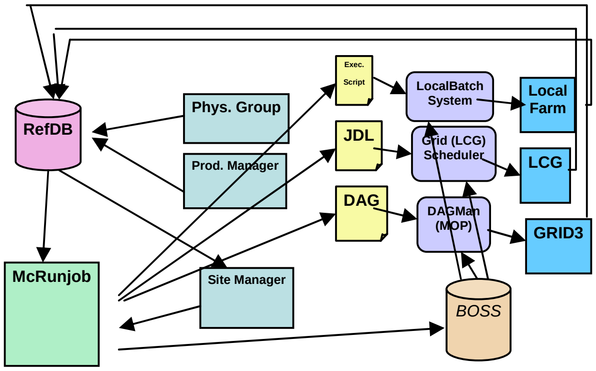

Figure 2 : Overview of the CMS production cycle

Site managers are notified by email that a request is assigned to their centre, they start a McRunjob session. All information required for jobs creation is fetched from RefDB to McRunjob through http. Jobs are submitted either to the local batch system or to the Grid.

Successful jobs send meta information to CERN by email to the allocated mail server. The UpdateRefDB script checks the mail box, performs job validation task, comparing META information sent by job with control parameters recorded in RefDB, in the case of a the successful check job gets validated in RefDB, and status of the corresponding assignment is updated.

CMS REFERENCE DATA BASE

RefDB represents a core part of the CMS production system. It has multiple functionalities and multiple clients both users and applications.

It is used for recording of the production requests and following of the request processing. Physicists or production managers can browse RefDB for getting information about already-registered applications, executables, datacards and can insert new ones. They get information about production progress of a given assignment or general production statistics on the level of one regional centre or for the full CMS production scale.

RefDB is used for distribution of work between different sites. It provides a main source of production instructions for the work-flow planner, McRunjob. It contains production templates for all processing steps, job-monitoring components for BOSS, definition of the detector and magnetic field geometries, description of the requests for creating of DAR distributions, job-level meta information required to create input for any production step. The number of jobs and events per job is also determined by RefDB

RefDB records book-keeping and metadata information of the produced data collections. Users can get a list of available data collections produced CMS-wide, or get the status of the collections which are still under processing.

MCRUNJOB

McRunjob provides a modular framework for

- creating batches of production jobs, possibly combining several processing steps in one job
- submitting them to different types of environment, such as various batch systems or Grids
- following the progress of job processing through a tracking directory
- publishing of the processed data collections :
  - creating of the POOL XML catalog for the collection chain
  - updating initialized COBRA META files with produced collection data

Meta information for every job contains XML catalog fragment and attributes required to make COBRA META files aware about data files produced by a given job. This information is recorded in RefDB and is recalled from RefDB by McRunjob during a separate publishing step. The XML catalogue for a given collection is composed from the catalogue fragments related to the collection jobs and data files belonging to the collection are “attached” to the collection COBRA META files.

BOSS

BOSS interfaces to a scheduler through a set of scripts developed for a local batch system or for a given Grid flavor. The job is submitted to a queue with a BOSS wrapper and is equipped to filter the standard output and standard error of the job with the user-defined filter.

In the filtering script a user can define the patterns he would like to trace and also possibly some actions which have to be implemented when a given pattern is found in the standard output. As soon as a predefined pattern is found in the standard output, extracted information is recorded in the BOSS data base. Users can browse the data base to follow job processing , for example one can find out how many events had been processed by any given job at any given moment.

Filters for the production jobs for all production steps have been developed by the production team and are recorded in RefDB. They are extracted from RefDB by McRunjob during job creation.

DATA FORMATS AND APPLICATIONS USED FOR RECONSTRUCTION

CMS reconstruction is done with the ORCA [12] application (Object Oriented Reconstructed for CMS Analysis) which in turn uses the CMS COBRA framework .

Digitization represents simulation of the DAQ process, its output is called “Digis”. Digi files are used as an input for reconstruction.

The output of the reconstruction step is called DST (data summary tapes). Reconstruction input and output files are both recorded in the POOL format.

For DC04 the functionality of ORCA was extended to store collections of reconstructed objects (DST) in multiple parallel streams.

DC04 RECONSTRUCTION

DC04 reconstruction was run at the Tier-0 (CERN).

Reconstruction code was released not long before DC04 started. There was no chance to test it at the production scale. Several ORCA versions, for bug fixes and functionality improvements were released during the 2 months of the data challenge. In total, 24 million events were reconstructed during DC04, but the same input data had been processed many times.

A reconstruction rate of 25 Hz corresponds to 2000 jobs per day using 100% of 500 CPUs. It was reached during limited periods of time, but not sustained. Running reconstruction at this rate did not create any problem, the bottleneck was not at the reconstruction level but at the level of further data distribution. Data distribution issues [13][14] are out of scope of the present paper.

Integration of the reconstruction into the production machinery was implemented very quickly. The most difficult part was to update the system for the “runStreams” executable, which implied producing multiple output collections from a single input collection. Current RefDB schema supports one-to-one correspondence between input and output collections, this caused a problem at the level of recording of job meta information in RefDB and at the level of further publishing. Development of the publishing procedure for output collections of the runStreams executable took about two weeks , but was developed in time, though the first publishing versions were rather time-consuming and required better automation.

Figure 3 demonstrates the data flow at the Tier-0 and is very similar to the one shown in figure 2. The main focus here is given to the preparation of data distribution.

Reconstruction input files were prestaged from CERN castor to the Input Buffer, from where they were copied to the worker node. Output data files (DSTs) were written to the castor pool which represented the Global Distribution Buffer (GDB). From the GDB files were transferred to the Tier-1s and were written to tapes. Files stayed in the GDB until they were recorded on the MSS at the remote sites, in which case they were set to “safe” in the Transfer Management Data Base (TMDB) [13]. Files having “safe” status in TMDB were cleaned from the GDB by a cleaning agent. Meta information of every reconstruction job was as usual recorded in RefDB, but at the same time it was copied to the Transfer Agent drop box.

The Transfer Agent used this information to update RLS, which kept information about all replicated files, and to update TMDB.

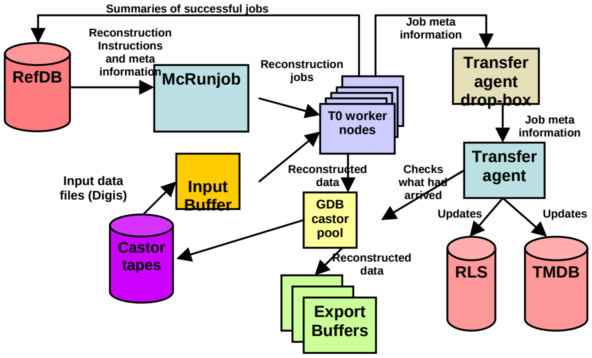

Figure 3: DC04 data flow at Tier-0

Job level meta information consisted of the pool XML fragment, which was used to update RLS, and of a file containing checksums of all data files produced by a given job. Information from the checksum file was used by the Transfer Agent to update TMDB for further transfer quality control. Every reconstruction job set up a "go" flag for the transfer agent to indicate that output of a given run is ready for distribution from Tier-0 to Tier-1s.

PUBLISHING OF THE RECONSTRUCTED DATA COLLECTIONS

As already mentioned, some problems were encountered while setting up the procedure for publishing of the data collections. They were caused by several different reasons described below.

- First version of "attachment" of COBRA meta files was very time consuming.
- COBRA meta files are not static, their content is changing as soon as more data gets attached to them, as a result they can not be distributed the same way as data files - using RLS, since RLS does not support versioning.
- Support of publishing in RefDB was developed for Objectivity and did not match the new persistency system.

All these problems delayed user access to the produced data collections and had a bad impact on the user analysis. For some of the problems temporary solutions were found during DC04. For example, COBRA META files had been distributed in the form of a zipped archive containing a whole set of meta files in one file with a flag reflecting status of a given version of META files.. This file could be registered in RLS without any ambiguity.

Most of the problems are addressed by the current development driven by the CMS production team.

The performance of the publishing procedure was improved and it was better automated.

A new user interface to RefDB for getting information about available data collections was developed following recommendations and requirements of the physics community.

A distributed system (PubDB) for publishing of XML POOL catalogues and COBRA meta files for produced and replicated data is currently under development. The first prototype of PubDB is already working at CERN, FZK, PIC, INFN. A new improved version is under development.

CONCLUSIONS

CMS production software has proven to be flexible and was very quickly updated for supporting DC04 reconstruction.

No serious problems were discovered in the part related to data processing at the given rate and in the production book-keeping system**.**

Ongoing development is focused on improving a system for publication to the CMS physics community information about available data collections and for distribution of meta information required for data access.

ACKNOWLEDGEMENTS

We would like to acknowledge Werner Jank, Nick Sinanis and IT division for the hardware infrastructure support and their immense help.

REFERENCES

[1] N. De Filippis et al., "*Tier-1 and Tier-2 real-time analysis experience in CMS DC04", ,* CHEP04, Interlaken, 2004

[2] A. Fanfani *, "Distributed Computing Grid Experience in CMS DC04",* CHEP04, Interlaken

[3] LCG2 in CMS , http://cmsdoc/cern.ch/cms/LCG/LCG-2

[4] CASTOR (CERN Advanced STORage manager), http://cern.ch/it-div-ds/HSM/CASTOR

[5] COBRA Project, http://cobra.home.cern.ch/cobra

[6] POOL Project, http://lcgapp.cern.ch/project/persist

[7] OSCAR Project, http://cmsdoc.cern.ch/oscar

[8] V. Lefebure, J. Andreeva *"RefDB: The Reference Database for CMS Monte Carlo Production"*, CHEP03, La Jolla, California, 2003

[9] G. Graham et al. *"McRunjob: A High Energy Physics Workflow Planner for Grid Production Processing"*, CHEP03, La Jolla, California, 2003

10] C. Grandi, A. Renzi , *"Object Based System for batch Job Submission and Monitoring (BOSS)"* CMS NOTE-2003/005

[11] DAR , http://computing.fnal.gov/cms/software

[12] ORCA Project, http://cmsdoc.cern.ch/orca

[13] T. Barrass et al., "*Software agents in data and workflow management*", CHEP04, Interlaken, 2004

[14] D. Bonacorsi et al., *"Role of Tier-0, Tier-1 and Tier-2 regional centres in CMS DC04,* CHEP04, Interlaken, 2004