CMS Conference Report

29 August 2005

The CMS analysis chain in a distributed environment

N. De Filippis¹⁾

T. Barrass²⁾, D. Bonacorsi³⁾, G. Ciraolo⁴⁾, M. Corvo^{5,3)}, G. Donvito⁶⁾, L. Faina⁷⁾, A. Fanfani⁸⁾, F. Fanzago⁹⁾, C. Grandi¹⁰⁾, V. Innocente⁵⁾, S. Lacaprara¹¹⁾, G. Maggi¹⁾, M. Maggi⁶⁾, A. Pierro⁶⁾, L. Silvestris⁶⁾, D. Spiga⁷⁾, L. Taylor¹²⁾, L. Tuura¹²⁾, T. Wildish¹³⁾

Abstract

The CMS collaboration is undertaking a big effort to define the analysis model and to develop software tools with the purpose of analyzing several millions of simulated and real data events by a large number of people in many geographically distributed sites. From the computing point of view, one of the most complex issues when doing remote analysis is the data discovery and access. Some software tools were developed in order to move data, make them available to the full international community and validate them for the subsequent analysis. The batch analysis processing is performed with workload management tools developed on purpose, which are mainly responsible for the job preparation and the job submission. The job monitoring and the output management are implemented as the last part of the analysis chain. Grid tools provided by the LCG project are evaluated to gain access to the data

¹⁾ Dipartimento Interateneo di Fisica dell'Università e del Politecnico di Bari e INFN Sezione di Bari, Bari, Italy

²⁾ University of Bristol, Bristol, United Kingdom

³⁾ INFN CNAF, Bologna, Italy

⁴⁾ Università di Firenze e INFN Sezione di Firenze, Sesto Fiorentino, Italy

⁵⁾ CERN, Geneva, Switzerland

⁶⁾ INFN Sezione di Bari, Bari, Italy

⁷⁾ Università di Perugia e INFN Sezione di Perugia, Perugia, Italy

⁸⁾ Università di Bologna e INFN Sezione di Bologna, Bologna, Italy

⁹⁾ INFN Sezione di Padova, Padova, Italy

¹⁰⁾ INFN Sezione di Bologna, Bologna, Italy

¹¹⁾ Università di Padova e INFN Sezione di Padova, Padova, Italy

¹²⁾ Northeastern University, Boston, United States of America

¹³⁾ Princeton University, Princeton, United States of America

and the resources by providing a user friendly interface to the physicists submitting the analysis jobs. An overview of the current implementation and of the interactions between the previous components of the CMS analysis system is presented in this work.

Presented at *X International Workshop on Advanced Computing and Analysis Techniques in Physics Research*,
Desy-Zeuthen, May 22-27, 2005

1 Introduction

The CMS experiment (Compact Muon Solenoid) [1] is a general purpose particle physics apparatus which will be located along the LHC accelerator (Large Hadron Collider) currently being built at CERN.

The detector is expected to start data taking in 2007 with a rate of about 150 Hz on disk which means tens of PetaBytes of data per year to be stored, moved and analyzed with many thousand of CPUs, by a large number of people in many geographically distributed sites.

In order to cope with large-scale computational and data access requirements and to ensure efficient data movement and reliable batch analysis processing, the CMS collaboration has defined the computing and the analysis model [2] and chose a distributed architecture based on some components of the Grid technology [3] to implement the analysis chain.

Grid tools provided by the European LCG (LHC Computing Grid) [4] and the American OSG (Open Science Grid) [5] projects, are evaluated to gain access to the data and the resources which are the most complex issues when doing remote analysis.

In the following section an overview of the flow of data between the CERN and the other CMS regional centres is given and a description of the tools used to perform different actions is provided. In section 3 the analysis workflow is detailed together with the main features of the tools used in a distributed environment. Job monitoring tools are introduced in Section 4.

2 CMS data flow

The logical flow of the CMS data can be easily introduced by giving firstly a short description of the current computing environment.

2.1 The computing environment

A hierarchy of computing regional centres, called Tiers, is defined in the CMS Computing Model. The system is geographically distributed and includes a single Tier-0 centre at CERN, a CMS Analysis Facility at CERN, several Tier-1 centres located at large regional computing centres, and many Tier-2 centres (smaller than Tier-1 centres but with substantial CPU resources).

Data collected by the CMS online data acquisition system are sent to the Tier-0 centre at CERN where raw data are archived, a prompt reconstruction is performed and a first version of the Analysis Object Data, AOD, is produced. All the high-level physics objects are stored in AOD together with the information sufficient to support typical analysis actions such as track refitting, ect.

Raw and first pass reconstructed events are distributed from the Tier-0 to a Tier-1 centre which takes custodial responsibility for those. AOD are also transferred to all Tier-1 centres that provide services for data archiving, re-processing, calibration, skimming and other data-intensive analysis tasks.

All AOD and a fraction of the first pass reconstructed events and RAW data are transferred to Tier-2 centres which provide capacity for iterative analysis, calibration activities and Monte Carlo simulation.

Additional computing resources are provided at Tier-3 centres that support the interactive use of CPUs and disks for local groups. A CMS Analysis Facility centre is also located at CERN in order to provide fast turnaround computing services local to the experiment.

CMS therefore needs to audit the creation and processing of data at any point in this environment; it needs to manage the movement of data through the system; it needs to validate data as it is created or arrives at a destination; and each site in the environment needs to publish the data it has available.

The auditing of data creation and processing is handled by a CMS dataset bookkeeping system; data location information is maintained at a high level by a dataset location system, while movement of data is managed by a dataset transport system; dataset validation is provided by a set of tools that run at each site; local dataset information is published into a local extension of the dataset bookkeeping and location systems.

A rough schema of the current CMS data flow is reported in Figure 1 and some details on how to perform data movement, validation and publishing are given in the following sections.

Figure 1: Data flow between Tier centers in the distributed architecture

2.2 Data distribution

The basic data collections that the CMS data management system is able to handle correspond both to the physicist abstractions for data samples, such as the "dataset" and the "event collection", as well as to the "packaging" concepts related to the computing system, such as files.

The "event collection" is the smallest unit that can be selected through the dataset bookkeeping system such as a run of real data from a particular trigger selection or a simulated run of Monte Carlo events. An "event collection" is made of physical files. The "dataset" is defined as any set of "event collections" that can be grouped and analysed together because of their physics attributes, like the trigger selection or the Monte Carlo physics generator, or due to the data format, like RAW or AOD data.

The CMS data movement is managed with a system named PhEDEx (Physics Experiment Data Export) [6]. The PhEDEx tool is a global system able to manage large-scale and prioritized transfers, O(1000) files per data samples; it is responsible of moving data from Tier-0 to Tier-1 and Tier-2 sites after different processing and for subsequent analysis; PhEDEx is also used to transfer the output files of the simulation and the analysis steps from Tier-2 sites to Tier-1.

PhEDEx uses an Oracle database at CERN to track and store the information about the transfer operation and it is based on the *gridftp* Grid component to copy files from different locations (also from disk to tape and from tape to disk).

PhEDEx is currently in production operation at all the Tier-1 sites and also at a number of Tier-2 sites.

2.3 Data validation

As soon as data files were transferred to a Tier site they are validated automatically as triggered by PhEDEx scripts or manually by the site manager in order to ensure their integrity and readiness for the end-user analysis. The set of procedures of data validation are currently implemented in the ValidationTools software [7] and consist of building local file catalogs by handling data and meta-data information, processing data via standard analysis executables and publishing validated information in a local database.

Different use cases are supported to validate both data samples officially produced by the collaboration, as well as private simulated events.

Different levels of checks of data consistency are also implemented according to the specific Tier-n needs. A time period of 120 msec per event is necessary on average for the full execution of validation of data samples consisting of simulated hits, digitized information of the CMS sub-detectors and pileup-events. As the validation procedure is time-expensive, a "Light" validation step is also implemented which is faster by a factor 10 with respect to the

deepest validation step; "Light" validation ensures internal consistency of files but not the integrity of events stored in them.

The ValidationTools software is in production at Italian Tier-1 (CNAF) and Tier-2 centres and also at several Tier-2/3s in Europe.

2.4 Data publishing

Currently bookkeeping and location services are provided by RefDB (Reference Database) [9], which is used to track Monte Carlo data; by PubDB (Publication Database) [8], which presents information about datasets available at a site; and by local file catalogues that map logical filenames to physical filenames.

The current prototype of the PubDB is a MySQL database where the local file catalogs properties and the validated information related to data in a Tier-n site (like the validation status, the total number of validated events, the run range) are published. That information are used by the workload management tools to handle analysis jobs.

The data validated and published in the local databases are expected to be ready and available to the analysis users for batch distributed processing.

3 CMS analysis work-flow

From the conceptual point of view the analysis work-flow is managed by the workload management system which has to provide tools for the large scale data access and processing; due to that the workload management system is expected to interact with Grid tools and services.

The workload management system provides the tools to translate the user request to analyze some data with a given application into a set of jobs to be run on the distributed computing system; the monitoring of that jobs is another task to be managed by the workload management system.

The basic requirement for the analysis job execution is the installation of the CMS software which has to be homogeneous and complete in all sites hosting data. The CMS environment to be created on remote machines should not differ from that which the analysis user actually finds on his/her desktop computer.

At present the main components of the CMS software are ORCA (Object-oriented Reconstruction for CMS Analysis) [10] that is the code for reconstruction and analysis, and COBRA (Coherent Object-oriented Base for Reconstruction, Analysis and simulation) [11] that is the software for the framework.

The CMS software installation is currently performed with the XCMSI tool (CMS Installation) [12] which does not require root privileges, allows save-able and reusable set-up, multi-platform support and includes validation procedures. In a Grid environment, mainly in LCG remote sites, the installation is managed by a CMS software manager by running installation Grid jobs.

The work-flow of the analysis chain in the LCG environment, as it is today, is shown in Figure 2.

The current workload management tool responsible for job preparation, splitting and submission is called CRAB (Cms Remote Analysis Builder) [13].

CRAB is a user-friendly tool to analyse data published in remote sites by using the Grid infrastructure but making Grid details mostly hidden to the user which is not supposed to be a Grid expert. CRAB has to be run by the analysis user on a local user interface machine where the user develops his/her code and that is configured to be able to submit jobs in remote sites via Grid services.

The user has to define the data sample name, the number of events to be analysed, the name of the executable of his private code and few other parameters via a configuration file. The executable, the libraries and the input files have to be provided locally and are sent in a compressed format in the Input Sandbox of the job.

The procedure of data discovery is performed querying both RefDB to derive the list of remote sites publishing the requested data and then PubDB to extract site specific information such as the available file catalogues and their access methods.

The job splitting is currently based on the number of events per job defined by the user in the configuration file, but it will evolve in a much more robust and user-configurable algorithm. The file catalogs as published in PubDB are downloaded and used by the analysis executable when the job reaches the worker node in a given site.

Figure 2: Work-flow for the CMS analysis chain in a distributed environment.

The jobs prepared in the previous steps are submitted from the User Interface to the Resource Broker that interprets the user requirements specified using the job description language [14]. The Resource Broker is responsible to find out the information about the available Grid resources (Computing Elements and Storage Elements) and drives the jobs to one or more selected sites hosting the data.

Job monitoring, automatic retrieval of the job output and final output management of the output files are also supported in CRAB.

The CRAB tool is actively used by several tens of CMS users all over the world. It is successfully used to access data at Tiers-1 centres and some Tier-2s, mainly in Italy. Currently a number of about $O(10^8)$ events were analysed with CRAB in the distributed infrastructure; only in the July and August 2005 $O(50000)$ analysis jobs were submitted with CRAB and ran over the Grid.

4 CMS job monitoring

The jobs submitted to the Grid can be monitored starting from the submission time using several tools.

The job monitoring, logging and bookkeeping can be performed by the BOSS tool (Batch Object Submission System) [15], which allows to deal with job-specific information, does not require to start any dedicated service on sites where the jobs run but it only requires access to a database for storing and retrieving job information. BOSS can be interfaced with most schedulers and is designed to work on computing farms.

Several sets of information can be stored in the BOSS database, some of them specific for CMS (like the number of analyzed events, the kind of analysis executable, etc.) and the others Grid related (like the exit code of the copy of the output files in a storage element).

BOSS was used for real-time analysis during the data challenge 2004 [16] and is currently used also to monitor Monte Carlo production jobs.

The monitoring in real-time of parameters internal to a C++ code or to a script is specifically addressed by another tool, JAM (Job Application Monitoring) [17], through a user-friendly interface.

The JAM tool allows to describe the user application with tags, to organize the applications in sets, to retrieve the values of the tags online. The information is stored in a pseudo-filesystem and rules on tags can be defined to classify the application as good/maybe/bad.

In the future plans JAM is expected to become a Grid service and distributed with the Grid middleware.

5 Conclusions

A CMS first working prototype for distributed user analysis is available and used by real users. CMS is using Grid infrastructure to access data and managing jobs for physics analyses.

Several software tools are under development, deployment and in production in many sites to implement the CMS analysis chain.

Several tens of users performed their analyses using CRAB and 100 millions events were already analysed successfully in that way.

6 Acknowledgements

We wish to thank all the people who provided feedback during the development and the deployment of the CMS analysis tools. Special thanks to V. Innocente for his always useful suggestions.

References

- [1] The CMS experiment: <http://cmsdoc.cern.ch>
- [2] "C-TDR - Computing Technical Design Report", **CERN/LHCC-2005-023**, 20 June 2005.
- [3] "The Grid: Blueprint for a New Computing Infrastructure", 2nd Edition, Morgan Kaufmann, 2004. ISBN: 1-55860-933-4.
- [4] LCG Project: <http://lcg.web.cern.ch/LCG> and "LCG Technical Design Report", **CERN-TDR-01 CERN-LHCC-2005-024**, June 2005.
- [5] OSG Project: <http://www.opensciencegrid.org> and R.Pordes et al. The Open Science Grid, presented at "Computing in High Energy and Nuclear Physics 2004" conference, Interlaken, Switzerland, 2004, in proceedings [Online] <http://www.chep2004.org>.
- [6] T. Barrass Software agents in data and workload management presented at "Computing in High Energy and Nuclear Physics 2004" conference, Interlaken, Switzerland, 2004, in proceedings [Online] <http://www.chep2004.org>.
- [7] N. De Filippis "Data Validation tools for CMS analysis", CMS Note 2005/XXX in preparation.
- [8] PubDB project: <http://cmsdoc.cern.ch/swdev/viewcvs/viewcvs.cgi/OCTOPUS/PubDB>
- [9] V. Lefebure, J. Andreeva RefDB: The Reference Database for CMS Monte Carlo Production, presented at Computing in High Energy and Nuclear Physics (CHEP) conference, La Jolla, California, 2003, in proceeding [Online] <http://www.slac.stanford.edu/econf/C0303241>
- [10] ORCA project: <http://cmsdoc.cern.ch/orca>
- [11] COBRA Project: <http://cobra.web.cern.ch/cobra> and V. Innocente COBRA - Coherent Object-oriented Base for Reconstruction, Analysis and simulation.
- [12] XCMSI project: <http://cern.ch/cms-xcmsi> and K. Rabbertz et al. "CMS software installation", presented at "Computing in High Energy and Nuclear Physics 2004" conference, Interlaken, Switzerland, 2004, in proceedings [Online] <http://www.chep2004.org>.
- [13] CRAB project: <http://cmsdoc.cern.ch/cms/ccs/wm/www/Crab>
- [14] "Job Description language HowTo", December 17th, 2001 available at <http://server11.infn.it/workload-grid/docs/DataGrid-01-TEN-0102-02-Document.pdf>
- [15] **CMS NOTE-2003/005**, C. Grandi, A. Renzi "Object Based System for Batch Job Submission and Monitoring (BOSS)".
- [16] **CMS CR 2004/048**, N. De Filippis et al. "Tier-1 and Tier-2 Real-time analysis in CMS DC04", presented at "Computing in High Energy and Nuclear Physics 2004" conference, Interlaken, Switzerland, 2004, in proceedings [Online] <http://www.chep2004.org>.
- [17] JAM Project: <http://jam.ba.infn.it>