September 17, 2001

Grid Infrastructure for Caltech CMS Production on Alliance Resources

V.Litvin[1], H. Newman[1]
S. Koranda[2], B.Loftis[2], J.Towns[2]
M. Livny[3], P.Couvares[3], T.Tannenbaum[3], J.Frey[3]

Abstract

The CMS experiment at the CERN Large Hadron Collider (LHC) will begin taking data in 2006. Along with the Petabytes of data expected to be collected, CMS will also generate an enormous amount of Monte Carlo simulation data. This paper describes solutions developed at Caltech to address the issue of controlled generation of large amounts of Monte Carlo simulation data using distributed Alliance resources. We will report on the results of production using Grid tools (Globus, Condor-G) to knit together resources from various institutions within Alliance in NPACI framework. The results of this effort will be discussed in addition to the design and implementation of the control system used.

[1] California Institute of Technology

[2] National Center for Supercomputer Applications

[3] University of Wisconsin - Madison

1 Introduction

The CMS experiment is a high energy physics experiment located at CERN, that will start data taking in the year 2006. The CMS detector (Figure 1) is one of the two general purpose detectors of the LHC accelerator. It is being designed and built, and will be used, by a world-wide collaboration, the CMS collaboration, that currently consists of some 1800 scientists in 144 institutes, divided over 31 countries.

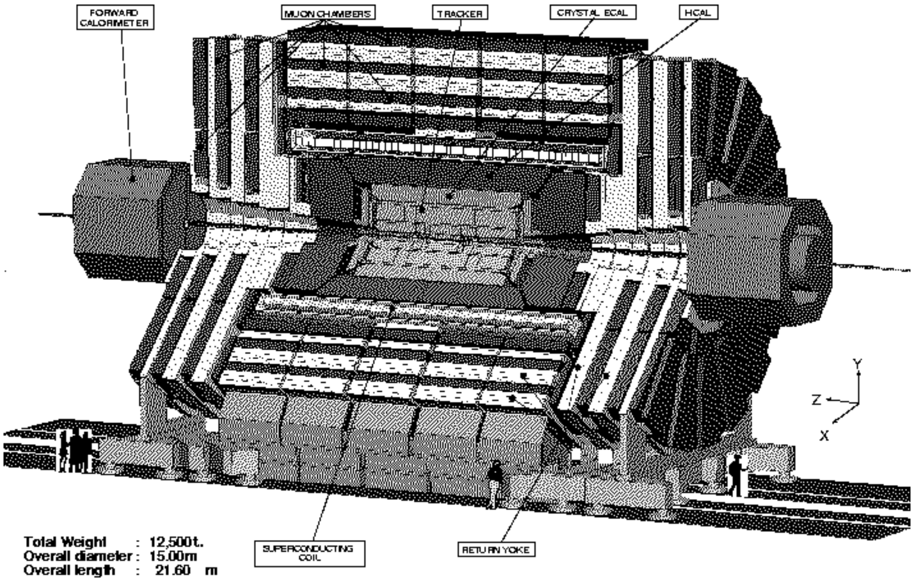

Figure 1: The CMS detector.

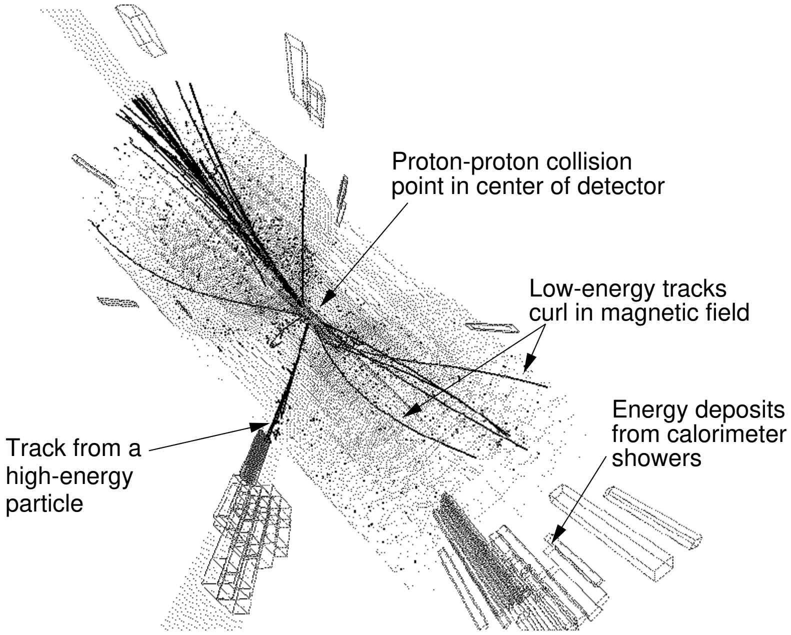

Figure 2: A CMS event (simulation at low luminosity).

In future operation, a large number of particle bunches, each containing some 10^{11} protons, circulate inside the LHC accelerator. At a rate of 40,000,000 times per second, the LHC accelerator lets two bunches of particles coming from opposite directions cross through each other inside the CMS detector. In every bunch crossing in the detector, on average 20 collisions occur between two protons from opposite bunches. The collision phenomena that occur during a single bunch crossing are called an *event*. Figure 2 shows an example of an event, with the

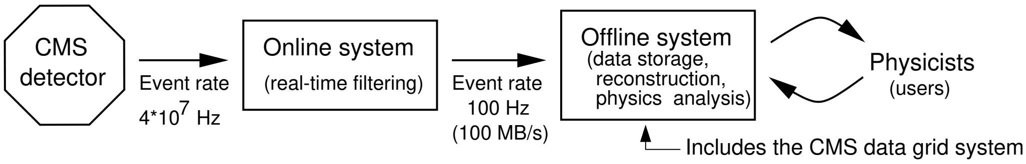

Figure 3: The CMS online and offline systems and their environment.

collision products emanating from the collision point. Note that the picture of the collision products is complex, and represents a lot of information: CMS has 15 million individual detector channels. The measurements of the event, done by the detector elements in the CMS detector, are called the *raw data*. The size of the raw data for a single CMS event varies around 1 MB.

CMS has the high event rate of 40,000,000 events per second because, like in many high energy physics experiments, the most important phenomena that the physicists want to observe will occur with only a very low probability in any single event. Of the 40,000,000 events in a second, some 100 are selected for storage and later analysis. This selection is done with a fast real-time filtering system (figure 3). The raw detector data of each selected event is stored as a set of 'data products'.

The nature of physics analysis implies several problems and opportunities for physics data handling systems like the CMS data grid. These problems and opportunities have been extensively studied in the RD45 project and elsewhere [30] [22] [23] [12] [26] [15] [25] [21] [27] [28] [16] [20]. Some of the main points are as follows.

The CMS offline hardware will consist of computer installations arranged in a hierarchical structure as shown in figure 4. This figure was adapted from [10]. In year 2007, the offline hardware consists of a large central computing facility at CERN, at tier 0, five 'regional centers' at tier 1, and about 25 still smaller centers at tier 2. Below that, there are institute servers at tier 3 and physicist's desktop workstations at tier 4. At the moment of writing, only a handful of sites are performing CMS regional center tasks. Going towards 2007, the number of deployed centers will slowly increase [13].

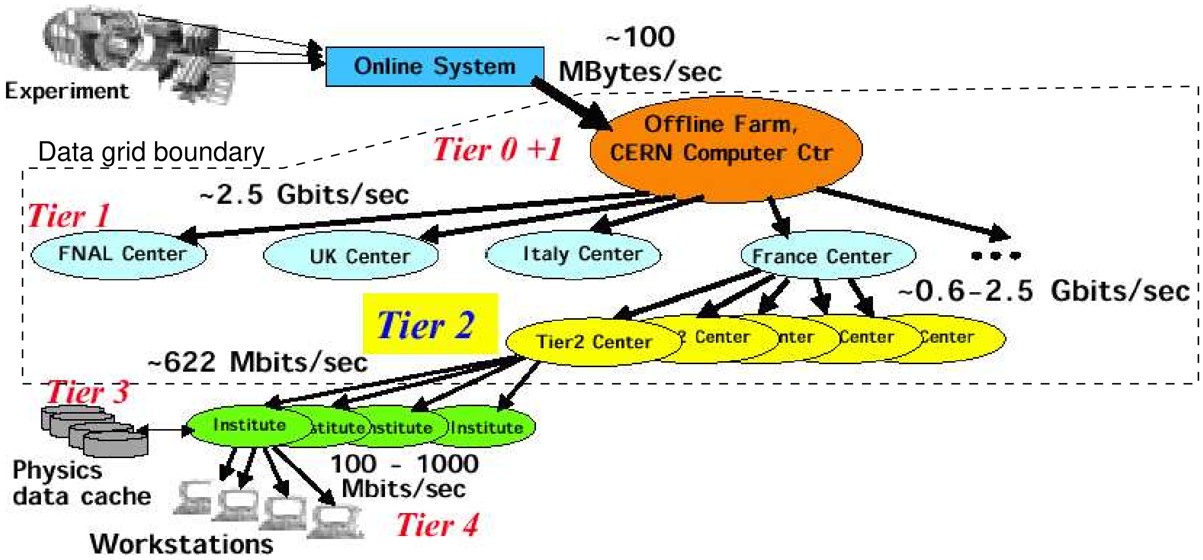

Figure 4: CMS hardware installations and network links (for 2007)

Around 2003, only some of the centers in figure 4 will already be in operation: the expected number is about 2 tier 1 centers and 6 tier 2 centers [13]. The scalability requirements for the grid components delivered to CMS by the end of 2003 are driven by the 20% data challenge milestone effort in 2004: the components need to scale to supporting a hardware configuration that has 20% of the projected 2007 capacity. The testbed configuration for the data challenge will exploit several shared resources in addition to the CMS-operated tier centers.

The **CMS data grid system** will support both CMS production and analysis and will seamlessly tie together resources in the tiers 0–2. The CMS data grid system is the most important system for the grid projects (GriPhyN, PPDG, and the EU DataGrid) in which CMS is involved. Whereas some other CMS software systems will also use grid technology, the CMS data grid system contains the grid components that should be targeted by the grid projects for which CMS is a ‘customer’.

2 Runs desciription

Someone pushed a button in Caltech to start the run. After the button was pushed necessary input files were automatically generated and using CondorG a significant number of CMSIM jobs were started on the Wisconsin Condor flock. Each of the CMSIM jobs produced a data file which was then automatically transferred to the UniTree mass storage at NCSA. After all the CMSIM jobs had finished at Wisconsin a job at NCSA was automatically started to begin the ORCA reconstruction phase of the run. Once the ORCA phase finished the output was automatically transferred to NCSA UniTree and the run was completed.

For the specialists here are the details for how we will accomplish this.

The Master Condor Dagman script looks like:

```
Job A /home/litvin/test90/simulation-test.sub DONE
Script POST A /home/litvin/test90/simulation-success.csh
Job B /home/litvin/test90/reconstruction-test.sub

PARENT A CHILD B
```

Each CMSIM job is really two jobs; two executables are required to run. For each of the CMSIM jobs then there is an ’A’ job and a ’B’ job. Further for each of the CMSIM jobs there is a ’pre’ script and a ’post’ script which is run. Condor DAGMan is used to run the pre script, then the A job, then the B job, and finally the post script for each of the CMSIM jobs. Condor DAGMan runs on the main Condor host at Wisconsin named beak.cs.wisc.edu, running under the “scheduler” universe (meaning that the pre and post scripts execute on beak as well as the DAGman executable which is managing the jobs).

To start Condor DAGMan executable on beak we will use CondorG running on a box at Caltech. We will submit a job to that local CondorG and specify the “globus” universe, with the gatekeeper being a special Globus gatekeeper on beak. This special gatekeeper on beak allows the executable to be run in the scheduler universe on beak. The executable specified in the job submitted to the local CondorG is simply the Condor DAGMan which needs to run on beak.

The DAGMan that runs on beak and which manages the CMSIM jobs requires a DAGMan DAG file, along with all the pre and post scripts and the Condor submit-description files for each individual CMSIM job. This collection of input files is sent to Wisconsin automatically by the CondorG at Caltech using Globus GASS.

The bottom line is that there will be no need to connect to Wisconsin and start the Condor run from there. CondorG from Caltech will effectively manage the whole process.

Each post script that gets run as part of each CMSIM job will run gsincftpput to transfer the generated data to NCSA UniTree. Authentication is automatic since the credentials have already been passed from Caltech to Wisconsin via CondorG and the Globus gatekeeper on beak.

Main part of the Condor Dagman script looks like:

```
universe        = globus
globusscheduler = beak.cs.wisc.edu/jobmanager-condor-INTEL-LINUX
environment     = CONDOR_UNIVERSE=scheduler

executable      = cmsimrun
arguments       = -f -t -l . -Debug 4 -Lockfile /shared/scratch2/litvin/Prod2001
/single_hgg_90/work/single_hgg_90.dag.lock  -Condorlog /scratch.local/cmsim/pers
onal_single_hgg_90.log -Dag /shared/scratch2/litvin/Prod2001/single_hgg_90/work/
single_hgg_90.dag -Rescue /shared/scratch2/litvin/Prod2001/single_hgg_90/work/si
ngle_hgg_90.dag.rescue -MaxJobs 100
```

```
input = single_hgg_90.tar.gz
remote_initialdir = /shared/scratch2/litvin/Prod2001

output          = simulation-test.out
error           = simulation-test.err
log             = condor.log

notification    = never

queue
```

When the last CMSIM job is finished at Wisconsin and the last data file has been moved to NCSA UniTree, the last job that the DAGMan will start is a globusrun to start the processing at NCSA. The globusrun will be sent to the gatekeeper on posic, the experimental Linux cluster at NCSA. The job that runs at NCSA will run in the "dtf" queue which has been setup on posic so that the job actually runs on the new 1GHz nodes that are part of the new NCSA platinum Linux cluster which will soon go into production. The last part of the job running at NCSA will again use gsincftpput to transfer the output to NCSA UniTree.

References

[1] http://www.griphyn.org/

[2] http://www.ppdg.net/

[3] http://www.eu-datagrid.org/

[4] http://cmsdoc.cern.ch/

[5] I. Foster, C. Kesselman, S. Tuecke. The Anatomy of the Grid: Enabling Scalable Virtual Organizations. To be published in Intl. J. Supercomputer Applications, 2001. Available at http://www.globus.org/research/papers.html#anatomy

[6] CAFE Group. Requirements from Physics Analysis Process to Software Architecture. http://cafe.web.cern.ch/cafe/

[7] S. Bethke et al. Report of the steering group of the LHC computing review. CERN/LHCC/2001-004, CERN/RRB-D 2001-3, 22 February 2001. Available from http://lhc-computing-review-public.web.cern.ch/lhc-computing-review-public/

[8] CMS collaboration. CMS Computing Technical Proposal. CERN/LHCC 96-45. December 1996.

[9] ORCA Tutorial at FERMILAB. http://cmsdoc.cern.ch/orca/FermilabTutorialWeek2000/

[10] P. Capiluppi. CMS World Wide Computing. Reported at the LHCC Comprehensive Review of CMS Software and Computing. October 2000. CERN. http://cmsdoc.cern.ch/cms/software/reviews/LHCC_review_oct_00/

[11] Monarc project web page. http://monarc.web.cern.ch/MONARC/

[12] K. Holtman, Prototyping of CMS Storage Management, Ph.D. thesis (proefontwerp), Eindhoven University of Technology, May 2000. see http://home.cern.ch/kholtman/ for a link to the pdf version.

[13] Ian Willers. CMS Interim Memorandum of Understanding: The Costs and How They are Calculated. CMS Note 2001/035.

[14] Koen Holtman. HEPGRID2001: A Model of a Virtual Data Grid Application. Proc. of HPCN Europe 2001, Amsterdam, p. 711-720, Springer LNCS 2110. Also CMS Conference Report 2001/006. Web site: http://kholtman.home.cern.ch/kholtman/hepgrid2001/

[15] K. Holtman, Clustering and Reclustering HEP Data in Object Databases, Proc. of CHEP '98, Chicago, USA. http://kholtman.home.cern.ch/kholtman/chep1/art1.web.html

[16] K. Holtman, H. Stockinger. Building a Large Location Table to Find Replicas of Physics Objects. CHEP'2000, Padova. http://kholtman.home.cern.ch/kholtman/olt_long.ps , http://chep2000.pd.infn.it/abst/abs_c074.htm

[17] K. Holtman, P. van der Stok, I. Willers. Automatic Reclustering of Objects in Very Large Databases for High Energy Physics, Proc. of IDEAS '98, Cardiff, UK, p. 132-140, IEEE 1998. http://kholtman.home.cern.ch/kholtman/ideas/autorecl_ideas98.html

[18] GDMP project web page. http://cmsdoc.cern.ch/cms/grid/

[19] Asad Samar, Heinz Stockinger. Grid Data Management Pilot (GDMP): A Tool for Wide Area Replication. IASTED International Conference on Applied Informatics (AI2001), Innsbruck, Austria, February 19-22, 2001.

[20] Heinz Stockinger, Asad Samar, Bill Allcock, Ian Foster, Koen Holt man, Brian Tierney. File and Object Replication in Data Grids, to appear in 10th IEEE Symposium on High Perfo rmance and Distributed Computing (HPDC2001) , San Francisco, California, August 7-9, 2001.

[21] K. Holtman, P. van der Stok, I. Willers. Towards Mass Storage Systems with Object Granularity. Proceedings of the Eighth NASA Goddard Conference on Mass Storage Systems and Technologies, Maryland, USA, March 27-30, 2000. http://kholtman.home.cern.ch/kholtman/mss.ps

[22] Using an object database and mass storage system for physics analysis. CERN/LHCC 97-9, The RD45 collaboration, 15 April 1997.

[23] Julian J. Bunn, Harvey B. Newman, Richard P. Wilkinson. Final Report of the Globally Interconnected Object Databases (The GIOD Project). October, 1999. http://pcbunn.cithep.caltech.edu/pubs/publications.html

[24] David Stickland. The Design, Implementation and Deployment of Functional Prototype OO Reconstruction Software for CMS. The ORCA project. CHEP'2000, Padova. http://chep2000.pd.infn.it/abst/abs_a108.htm

[25] M. Schaller. Reclustering of High Energy Physics Data. Proceedings of SSDBM'99, Cleveland, Ohio, July 28-30, 1999.

[26] M. Schaller. Persistent ATLAS Data Structures and Recluste ring of Event Data. Ph.D. thesis, Geneva, September 1999.

[27] A. Shoshani, L. Bernardo, H. Nordberg, D. Rotem, A. Sim. Multidimensional Indexing and Query Coordination for Tertiary Storage Management. Proceedings of SSDBM'99, Cleveland, Ohio, July 28-30, 1999.

[28] A. Hanushevsky. Developing Scalable High Performance Terabyte Distributed Databases. Proceedings of CHEP'98, Chicago, USA.

[29] http://cmsdoc.cern.ch/cms/production/www/html/general/inde x.html

[30] http://wwwinfo.cern.ch/asd/rd45/index.html

[31] http://www.globus.org/

[32] http://www.cs.wisc.edu/condor/

[33] http://cmsdoc.cern.ch/cgi-cmc/scrampage

[34] http://wwwinfo.cern.ch/asd/geant4/geant4.html

[35] http://anaphe.web.cern.ch/anaphe/