CMS Monte Carlo Status, Performance and Future plans

Hans Wenzel *

Fermi National Accelerator Laboratory, P.O.Box 500, Batavia, IL 60510-0500

Abstract. In this article we will discuss the current status of the CMS software. A first major Monte Carlo production was conducted in spring 2000. We will discuss the experiences we collected during this production. A Distributed Monte Carlo production is currently in progress (Fall 2000).

INTRODUCTION

CMS is an experiment under construction for the LHC pp collider at CERN. The LHC is designed to operate at $\sqrt{s}$= 14 TeV. The nominal luminosity of $10^{34} cm^{-2} s^{-1}$ and bunches crossing every 25 nsec result in 10^9 pp interactions per second. The first collisions are expected in 2005. The LHC will not commence with its nominal luminosity but a low-luminosity start-up phase of several years with 10% of the nominal value in the first year, 33% in the second year and 67% in the third year (1). LHC experiments will have to deal with larger data sets, more (and wider distributed) computing power with collaborators more spread out than ever[1] before experienced in high energy physics[2].

The first major Monte Carlo production was conducted in spring 2000. Distributed Monte Carlo production is happening right now (Fall 2000). The highest priority for the production effort was the Higher Level Trigger validation to be performed using full simulation and reconstruction of high luminosity LHC events. Other goals of the Monte Carlo production were to provide 'realistic' simulation data to study trigger and physics performance of CMS, to provide data in order to develop reconstruction and calibration algorithms and to provide feed back for realistic simulation of network/CPU usage and access patterns (MONARC)(2).

CMS relies on a distributed computing model with 1/3 of the computing and data storage at tier 0 (CERN) , 1/3 at tier 1 (CERN, Fermilab,INFN, IN2P3, ??) and 1/3 at tier 2 centers. For this to work entirely new tools must be developed and exercised. These includes tools for data base management, load balancing, data replication, monitoring, production farm management and production system replication. The current production efforts are the initial step to develop and use these tools and to estimate if the computing model will scale up to the size and complexity expected for LHC experiments. The major challenges for LHC software and computing today include:

- Events are big (raw event is 2MB).
- at the design luminosity of $10^{34} cm^{-2} s^{-1}$ approximately 17 minimum bias events per crossing are produced. With the bunch crossing time so short (25 nsec) a realistic detector digitization has to take into account the fact that events from different bunch crossings contribute to the digitization. At least 9 crossings (-5 to +3) contribute to calorimetry digitization [3]. Typically information from more than 150 (9×17) minimum bias events is needed for each signal event. Thus for 1 million signal events you would need to generate > 150 million minimum bias events. This is impossible with current available CPU, storage etc.. Instead we include the minimum bias events in the digitization step and recycle simulated minimum bias events. For that we created a minimum bias data set consisting of a few hundred thousand events. For each signal event we randomly select the necessary number of minimum bias events in a straightforward manner. Problems can arise when one single minimum bias event by itself would trigger the detector. One would get this trigger many many times. Therefore it is necessary to filter the minimum bias events, but remember to take into account the removed events. The size of the 150 required minimum bias events which have to be read in for each signal event is greater than 50MB. This results in a massive data movement problem. The

∗ representing the CMS collaboration

[1] 1700 physicists of 150 institutes in 32 countries

[2] there are two possible attitudes: optimistic- great opportunity or pessimistic- the greatest computing nightmare ever experienced!

[3] Muon digitization is affected by even more crossings

advantage of this approach is that it is easy to study the same signal events at different luminosities and also with and without pile-up.

- The CMS Tracker is immersed in a high magnetic filed of 4 Tesla. The tracks of low momentum charged particles loop in the magnetic field and can persist for many crossings. This requires track finding in a very complex environment.

- The total tracker material adds up to 1 radiation length resulting in lots of bremsstrahlung for the electrons. This makes matching tracks to calorimeter clusters a non-trivial task.

One CMS event requires of order 100-400 times more computing to digitize than a Tevatron Run II event.

CMS SOFTWARE

CMS decided to use object oriented technology and C++ has been chosen as the programming language. This allows the use of many modern C++ standard tools (e.g. standard library) (3). The Software and runtime environment is managed and distributed using SCRAM [4]. The source code is organized into modules using CVS in order to support a multi-developer environment. Currently the Monte-Carlo events generators and the detector simulation are still written in Fortran.

The Framework

The basic framework is called CARF[5]. To achieve maximum flexibility CARF implements an "implicit invocation architecture" (Action on Demand) (4) where Modules register themselves at creation time and are invoked when required. In this way only those modules really needed are loaded and executed. CARF pins Database objects (Objectivity) in memory for the duration of an event. The user accesses C++ objects and for the user it is invisible if the object is transient or in the Database. No "Objectivity handles" appear in the user code. All modules see the same persistent data and each modules can change its local copy of the data, but one can not unintentionally change persistent data nor the data which is viewed by another module in the same reconstruction job.

[4] **S**oftware **C**onfiguration, **R**elease **A**nd **M**anagement.
[5] **C**MS **A**nalysis and **R**econstruction **F**ramework

Monte Carlo generators and Detector Simulation

There are several physics generators available in the CMS software environment. The current production (fall 2000) uses Pythia (version 6.152), Isajet (version 7.48) and in some cases the TAUOLA package for correct tau decay dynamics. The results of the generators are stored as HEPEVT ntuples.

For the simulation of the CMS detector response we use CMSIM (version 120) (5) which is written in FORTRAN and based on GEANT3. The plan is to retire this package, as soon as OSCAR[6] is available and is sufficiently tested. OSCAR is written in C++ and based on GEANT4 at this time all sub-detector geometries are in and OSCAR is now being interfaced to CARF to use persistent storage. FAMOS is a fast parameterized simulation written in C++ and fully integrated into CARF. A first release is underway. FAMOS is required for the Physics TDR II planned for 2003. A fast simulation ,for example, will allow us to explore SUSY parameter space.

Reconstruction

The reconstruction framework called ORCA [7] is written in C++ and is fully embedded in the CARF framework. ORCA provides a sub-detector reconstruction skeleton into which new developers can implement realistic objects and algorithms which can be used for final detector optimizations, trigger studies or global detector performance evaluation. The ORCA sub-systems relevant for production include:

- **OOHits:** is the first step of OO reconstruction. OOHits reads CMSIM files which are in Zebra format and writes the simulated hits into an Objectivity database. This process is completely IO limited and does not require a lot of CPU.

- **OODigis:** adds minimum bias events (> 150) of several crossings. The Pile-up is selected randomly from a sufficiently large database of minimum bias events. OODigis then performs the full digitization and stores the resulting objects in Objectivity.

- **RecReader:** this is where the final steps of reconstruction is done including: calorimetric clustering, jet finding, muon segment finding, track finding and primary vertices finding using information from the silicon pixel detectors.

[6] **O**bject oriented **S**imulation for **C**MS **A**nalysis and **R**econstruction
[7] **O**bject **R**econstruction for **C**MS **A**nalysis

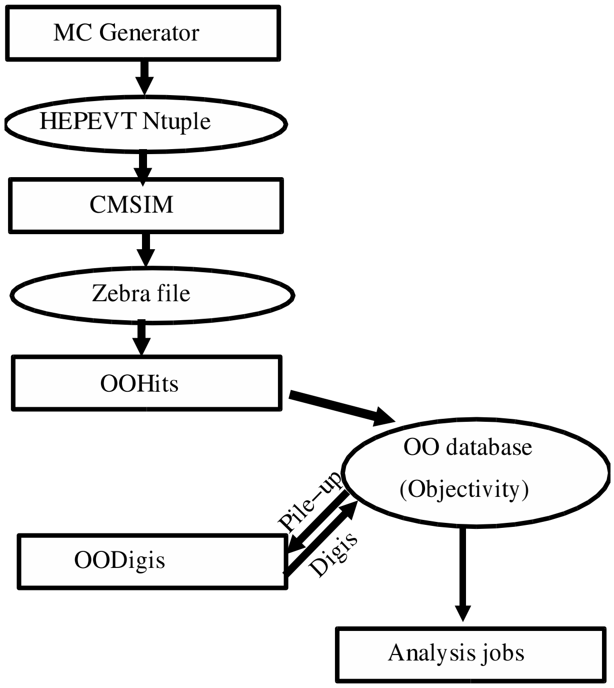

FIGURE 1. Steps and datasets involved in a CMS production job.

SPRING 2000 PRODUCTION

The first major Monte Carlo production was conducted in spring 2000. The steps involved are shown in Figure 1. The first steps of production (MC generation and the simulation) were partly done in regional centers outside of CERN. The Zebra files of the simulation were then transfered to CERN. The OO reconstruction (OOHits and OODigis) was done at CERN. The OO processing was done on a Linux farm. 70 CPU's were dedicated to projects of the muon physics group. Typically these jobs required 90 seconds/event and resulted in 20MB/event output. An additional 70 CPU's were dedicated to projects of the jets and missing E_t physics group with typically 60 seconds/event and 35MB/event. 30 Linux computers were used as dedicated database servers (running the Objectivity AMS server) to deliver the minimum bias events to the digitization processes. The results were written to a Objectivity database on a high speed SUN server. The database access patterns were similar to those expected for an analysis facility with 140 jobs reading asynchronously and chaotically from 30 servers writing into one server. The best reading rate achieved out of Objectivity was $\approx$70MB/sec and a continuous 50MB/sec reading rate could be sustained.

There was not enough disk space to keep the entire data base, nor even the minimum bias sample on disk therefore non disk-resident data needed to be staged from tape (MSS). OOHits was completely limited by the IO performance into and out of MSS, OODigis was IO limited in reading Pile-up events from dedicated database servers. Analysis jobs by physicists are again limited by IO performance staging data in and out of MSS. This is due to limited amount disk space (the ratio Tape/Disk=6).

FALL 2000 PRODUCTION

Production started in October 2000 and is still ongoing. Eight regional centers in Europe and the US are participating. Most centers do the only generation and simulation steps and transfer the resulting Zebra files to CERN using Perl scripts using scp. Some of the regional centers (e.g. Fermilab) are in the process will of setting up their production farms to do OO reconstruction (OOHits and OODigis). To move the Objectivity databases to and from CERN, the GDMP tools are used (6)

NEAR FUTURE

After we finish the current production (Fall 2000), a series of milestones with ever increasing functionality has to be met. To answer the fundamental questions of the collaboration on trigger and physics performance it is necessary to develop and test many new features in the computing and software. Some of the goals include:

- The use of the production data to develop algorithms and trigger strategies to achieve a final factor of 10 reduction in level 2 trigger rates. The results will be included in the final Trigger/DAQ TDR due in spring 2001.
- In 2002 a computing and software TDR and a 5% [8] data challenge are planned.
- In 2003 we plan to have the final Physics TDR.
- In 2004 we plan to perform a 20% data challenge.

[8] Here the percentage numbers refer to the level of complexity not necessarily the total amount of data.

REFERENCES

1. *The Tracker Project, Technical Design Report*, CERN/LHCC 98-6, CMS TDR 5.
2. http://monarc.web.cern.ch/MONARC.
3. http://cmsdoc.cern.ch/cmsoo/cmsoo.html.
4. M. Shaw *Some Patterns for Software Architecture*, Proceedings of the Second Workshop on Pattern anguages For Programming, Addison- Wesley, Reading Masachusetts, 1994
5. http://cmsdoc.cern.ch/cmsim/cmsim.html
6. http://www.globus.org.