CMS Full Simulation for Run-2

M Hildreth[1,2], V N Ivanchenko[†,3,4,5], D J Lange[6] and M J Kortelainen[3]

On behalf of the CMS Collaboration

[1] Université de Notre Dame du Lac, Notre Dame, IN, USA
[2] FNAL, Batavia, IL, USA
[3] CERN, CH1211 Geneva 23, Switzerland
[4] Ecoanalytica, 119899 Moscow, Russia
[5] Geant4 Associates International Ltd, United Kingdom
[6] Lawrence Livermore National Laboratory, Livermore, CA, USA

Abstract. During LHC shutdown between run-1 and run-2 intensive developments were carried out to improve performance of CMS simulation. For physics improvements migration from Geant4 9.4p03 to Geant4 10.0p02 has been performed. CPU performance has been improved by introduction of the Russian roulette method inside CMS calorimeters, optimization of CMS simulation sub-libraries, and usage of statics build of the simulation executable. As a result of these efforts, CMS simulation has been speeded up by about factor two. In this work we provide description of updates for different software components of CMS simulation. Development of a multi-threaded (MT) simulation approach for CMS will be also discuss.

1. Introduction

The CMS full simulation is based on the Geant4 toolkit [1], [2]. It has benefitted from many years of effort on detailed descriptions of the CMS geometry and detector response [3]. In particular, the performance of the CMS calorimeter simulation was studied with comparisons to test-beam data. Because of this, QGSP_FTFP_BERT_EML Geant4 Physics List has been established that is optimal for CMS [4]. For data analysis it is essential to use a stable version of the simulation, so for the CMS Monte Carlo production in 2012, Geant4 version 9.4p03 with a few CMS private patches was used. The total amount of Monte Carlo events produced for CMS in 2012 is about 6.5 billion (the total for the 7 and 8 TeV running is about 10 billion). This version of the simulation is used for legacy Monte Carlo productions for both the 7 TeV and 8 TeV data analyses.

In the run-2 at 13 TeV, we expect a substantially larger dataset, higher particle multiplicity, and higher pileup. With no changes to the CMS software (CMSSW) simulation framework, it was estimated that the typical simulation time per event would be about 25% more than that of the 8 TeV productions. This challenge for the CMS software required an increase in speed of the Monte Carlo production by a significant factor without compromising physics performance. During large shutdown of LHC intensive developments were carried out for various aspects of CMS simulation and several important improvements have been introduced:

† Corresponding author Vladimir.Ivantchenko@cern.ch

- usage of the new version of Geant4 10.0p02;
- optimisation of CMSSW code;
- introduction of the Russian roulette method;
- usage of statics build of the simulation executable;
- the MT simulation framework.

The speedup factor two was achieved for the simulation step of data processing as a total result. Below we will describe the most important aspects of these developments. The descriptions of other CMS activities for the run-2 simulation are outside this paper, namely: improved CMS geometry [5], new approach for pile-up overlay [6], premixing of pile-up simulated events [7], and fast simulation of the CMS experiment [8].

2. CMS Full Simulation evolution

As mentioned above, the legacy version of the CMS simulation provides good accuracy in the simulation predictions. However, there are improvements in several aspects of the Geant4 electromagnetic and hadronic physics that are important for future high statistics analyses. The CMS strategy is to adopt each new Geant4 version into the CMSSW development branch. Normally, switching to the new Geant4 version should require intensive validation efforts if done "from scratch" against data. However, the legacy version is well-validated versus real data, so the initial validation is performed between simulation results obtained with different CMSSW configurations which use different Geant4 versions. This type of validation is being done on a regular basis for all sub-detector signals and for all important physics observables. For calorimeters comparisons with the test-beam data were also performed.

In order to monitor the influence of changes to Geant4 on the CMS calorimeter response and resolution, simulations of proton and pion responses in the combined electromagnetic (ECAL) and hadronic (HCAL) calorimeters test-beam setup have been done for different Geant4 Physics Lists and different Geant4 versions. For purposed of comparison, the normalisation in each case is performed using the response of the calorimeter to a 50 GeV electron beam. Since Geant4 version 9.4 models of multiple scattering were upgraded and the evolution was continued until Geant4 version 10.0 [9]. As a result, the test beam simulation predicts a higher electron response in HCAL. Due to this fact the visible energy for hadrons is reduced by about 4%. In order to keep the hadronic response unchanged between Geant4 versions, the old Urban93 model is currently configured in the default CMS Physics List for electrons and positrons below 100 MeV. All other electromagnetic and hadronic models are used from the Geant4 distribution.

The usual scheme of CMS simulation upgrades has been to adopt new Geant4 versions and feedback any observed issues to the Geant4 Collaboration for incorporation into future releases rather than to apply private patches. In preview of the run-2 this strategy was slightly modified. Significant efforts have been committed to study the CPU performance of the development Geant4 version 9.6p02 and the list of updates for the new version 10.0 has been proposed to the Geant4 Collaboration. As a result of all efforts, Geant4 10.0p02 was adopted for the run-2 production and is carrying out for few billion Monte Carlo events production in 2015.

For CPU and memory profiling the IgProf tool was used for detailed profile analysis of the simulation step. Three characteristic types of events produced by PYTHIA were studied: *minimum bias*, $Z \rightarrow e^+e^-$, and *t-tbar*. It was observed that the profiling results for Geant4 classes for different primary events are compatible for 8 GeV, 13 GeV, and for all types of events. At the same time, contribution from CMSSW specific computations significantly increased for 13 TeV runs. To improve this situation a complete review has been done of CMSSW classes used in simulation (user actions and Monte Carlo truth handling). As a result, the hot spots were removed and CPU performance improved. The most important change was in substitution of loops over *G4LogicalVolume* vectors by loops over *G4Regions* in *SteppingAction* and *StackingAction* classes.

Repackage of all shared libraries in CMSSW that depend on Geant4 into a single static library to "hide" Geant4 from the rest of CMSSW has been implemented. Geant4 library is also statically built. This allows to more aggressively optimize executable at link time adding

-flto –Wl,--exclude-libs,ALL

This method provides ~10% performance gain. Impact on simulation code developers is minimized by keeping shared libraries cached in a release. Static library rebuild is the only extra step if a developer builds a package in this static library. Worse to note, that an extension of this approach to the digitisation and reconstruction steps of data processing is difficult, because for the simulation number of merged libraries is limited (~20), which is only 2% of total number of CMS libraries.

3. Russian roulette method

There are several options to improve the CPU performance of any simulation. First of all, it is possible to optimise Geant4 configuration for a particular use case. Continuous efforts of the Geant4 Collaboration are spent to speedup of Geant4 itself. Another approach is to modify the implementation of Geant4 in the CMS framework, either by changing the methods by which particle showers are described, or changing the manner in which Geant4 deposits the particle energies.

An example of the first modification, merely mentioned in passing here, is to use GFlash parameterisations of high energy showers instead of a detailed simulation [4]. This has been previously implemented in the CMS simulation framework, but has only been adopted to date in the simulation of far-forward detectors.

A different modification is to optimise the usage of Geant4 specifically within CMS, using modifications of Geant4 operating parameters. This was done in the past [3], [4] when the CMS simulation was established for the first experimental run using the QGSP_FTFP_BERT_EML Physics List as the default, based on comparisons with test beam data.

Further improvements may require a fresh view on how Monte Carlo simulation is performed. Specifically, the Russian roulette (RR) method, well known in neutron physics [10], [11], is a promising candidate for a further investigation. The method itself is simple: only part of the secondary neutrons are transported by a Monte Carlo engine but the signal of these transported neutrons is multiplied by a factor inversely proportional to the fraction of neutrons that are randomly killed. Applying the RR method can provide a valuable savings of CPU time. The mean value of the energy deposition is predicted with very high accuracy, but second and higher moments of the signal may be biased. On top of fluctuation introduced by the RR method intrinsic resolution of a calorimeter and detector response effects are applied. Final resolution is a convolution of all factors. So, simulation may predict detector response with desired accuracy but adoption of the RR technique requires substantial validation. In the case of CMS, studies have shown that there are many neutrons and gammas produced in the CMS calorimeters and the RR method can be applied to reduce low energy part of the spectra (figure 1).

The effect on CPU for other frequently produced particles, such as electrons or protons, is much less and application of RR to them is not useful. Upper limits on energy for the application of the RR method were defined using the information in figure 1: the upper limits proposed are 10 MeV for neutrons and 5 MeV for gammas. These upper limits are needed because application of RR to energetic particles may significantly bias final simulated signal.

The effect of the RR method was first studied using a stand-alone simplified calorimeter setup representing the CMS electromagnetic calorimeter (ECAL) and the hadronic calorimeter (HCAL) (figure 2). The RR method for this setup with a high-energy incident pion beam provides a significant CPU speedup with statistically similar detector response. For low-energy pions, such as the 1 GeV case shown, there are some variations of the signal shape. However, for LHC experiments each of these pions is usually a part of a hadronic jet, so some variation of single hadron response is acceptable. The CPU savings for applying RR to neutrons is about 40%, to gammas about 6%. Due to this success, the RR method was implemented for CMSSW as an option which is configured in the CMSSW run-time python script. It may be enabled in the following CMS regions (here region means *G4Region* [2]):

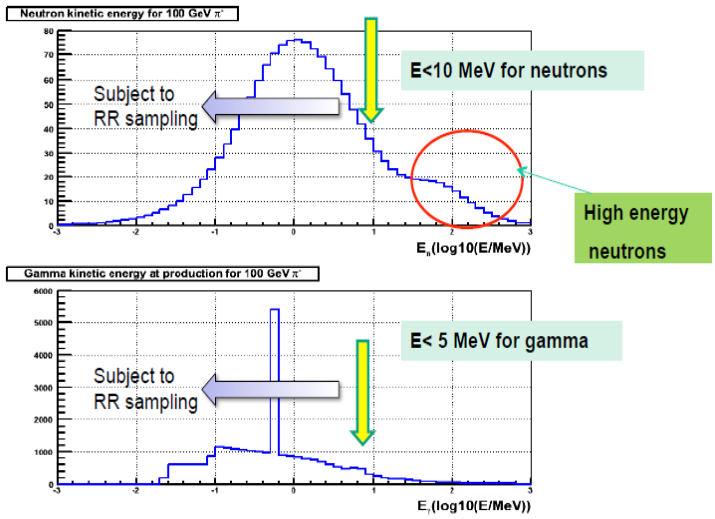

Figure 1. Energy spectra of neutrons (top) and gammas (bottom) for 100 GeV primary π- in CMS calorimeters. The upper energies for the RR suppression are 10 MeV for neutrons and 5 MeV for gammas.

- Pre-shower;
- ECAL;
- HCAL;
- Iron York;
- Castor;
- World.

The World region means all periphery geometry components which are not included into any other region. For each such region the RR factor *F* is defined separately for neutron and gammas. This factor is the probability for neutrons or gammas to be tracked further: before adding the track to the secondary particle stack of Geant4 a random number is checked to decide if to continue this track or to kill it. If track is continued its weight is set to *1/F*; the weight is propagated to all secondaries of this particle and its energy deposition in sensitive volumes of the calorimeters is multiplied by the weight. As a result, energy depositions in ECAL and HCAL with and without the RR method are the same.

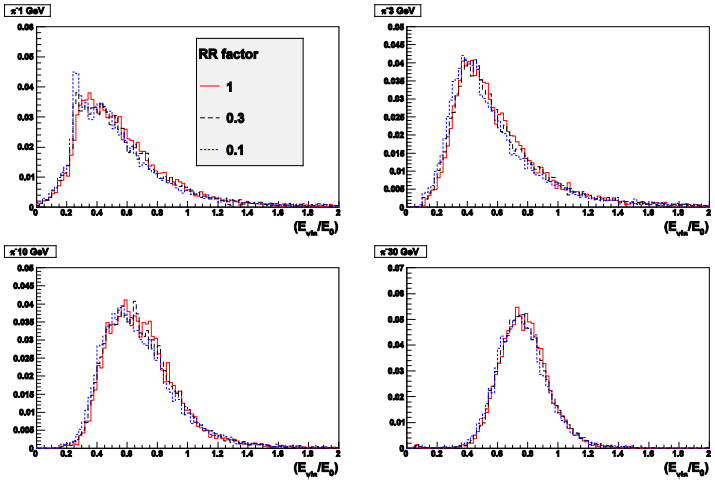

Figure 2. Energy response of a stand-alone combined (ECAL + HCAL) calorimeter for high energy pion beams with and without RR enabled: red histogram – no RR, F = 0.3 – black dashed, F = 0.1 – blue dotted.

The dependence of CPU on RR factors is shown in figure 3 for three types of event in CMS. When the RR factor is decreased too much, there is naturally no longer CPU advantage since an overwhelming fraction of low-energy particles have already been killed. Using these results, RR factor values were selected to be 0.1 for neutrons and 0.3 for gammas. The next step in the adopting of the RR method was to perform a physics validation of the CMS full simulation with RR enabled.

The validation for full CMSSW was carried out using the standard CMS validation tool. Results with and without RR enabled are in good agreement for all subsystems and all checked distributions except the width of reconstructed gamma versus Monte Carlo truth in the ECAL barrel. The RR method introduces extra smearing which is not seen in the ECAL endcap or in HCAL but is clearly visible in the ECAL barrel due to its higher intrinsic energy resolution. Because of the importance of an accurate simulation of isolated high energy gammas an extra limitation to the RR method has been added: if a secondary gamma is produced in the ECAL or pre-shower regions and if its parent is a gamma, electron, or positron, then the RR suppression is not applied. This final configuration of RR has been adopted for the development version of CMSSW. It currently provides about a 30% speedup of the full simulation in production mode.

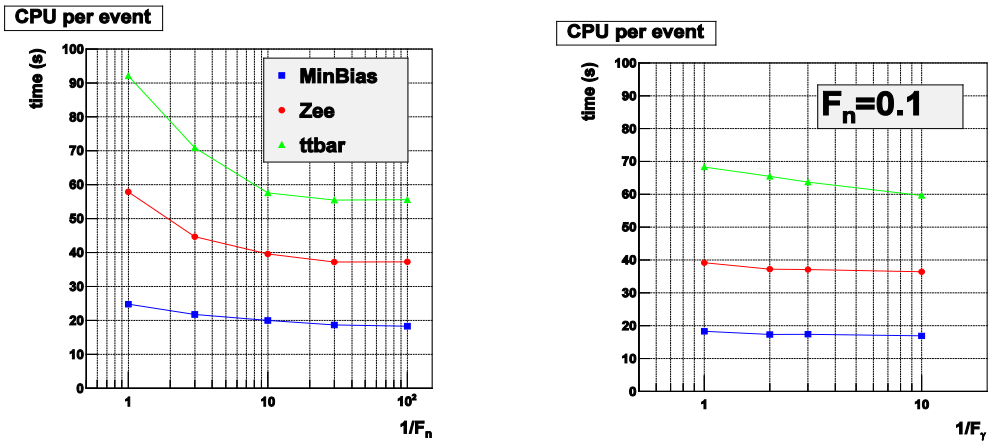

Figure 3. CPU per event for different event types as a function of inverse RR factor: RR for neutrons – left; for gamma – right. Geant4 version 9.6p02 is used.

4. CMS simulation performance for run-2

As a result of all improvements the CMS simulation step of the data processing for run-2 becomes about two times faster than that of run-1. Main contributions to this improvement are

- upgrade to Geant4 10.1p02 (~5%);
- CMSSW code optimisation (~15%);
- implementation of the RR technique (~30%);
- the library repackaging method (~10%).

IgProf profiling of the simulation step for the CMS production allows to identify contributions from different components of simulation (figure 4). The current leading contribution comes from tracking of particles inside CMS.

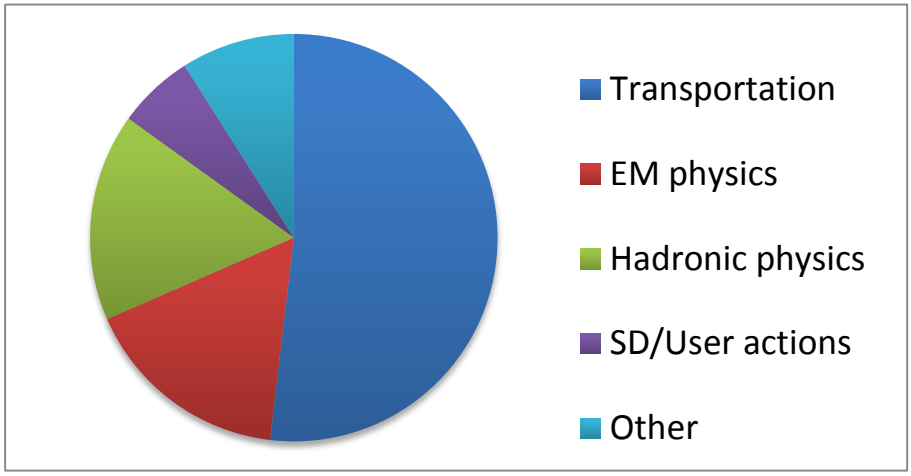

Figure 4. The diagram of CPU time distribution between different components of CMS simulation.

5. Development of CMS multi-threaded simulation

Geant4 version 10.0 was chosen by CMS also because it is multi-threaded capable [12]. This allows development of multi-threaded version of CMS simulation in parallel with the mainstream sequential production version. This become possible, because for CMSSW a general multi-threaded framework has been developed [13].

For the CMS multi-threaded simulations additional modules and manager classes were added (figure 5). This is needed because the threading models in CMSSW and Geant4 are different. CMSSW employs task-based parallelism (TBB) and lets TBB to schedule the tasks to the threads, while Geant4 uses worker threads explicitly and thread-local storage to hold the thread-private data. In addition, Geant4 uses the thread the program was started with as a "master thread" to do the global initialisation. This master thread cannot be used as a worker thread. On the contrary, in CMSSW all threads are equal and independently perform data processing.

These differences imposed certain challenges for integrating Geant4 MT into CMSSW. The CMSSW module running multi-threaded Geant4 (*OscarMTProducer* in figure 5) was implemented as a "stream" module, i.e. there is one object instance of it per thread. The first module object starting to execute launches a new *std::thread*, outside of TBB control, and runs the Geant4 global initialisation in that thread, i.e. it acts as the "master thread".

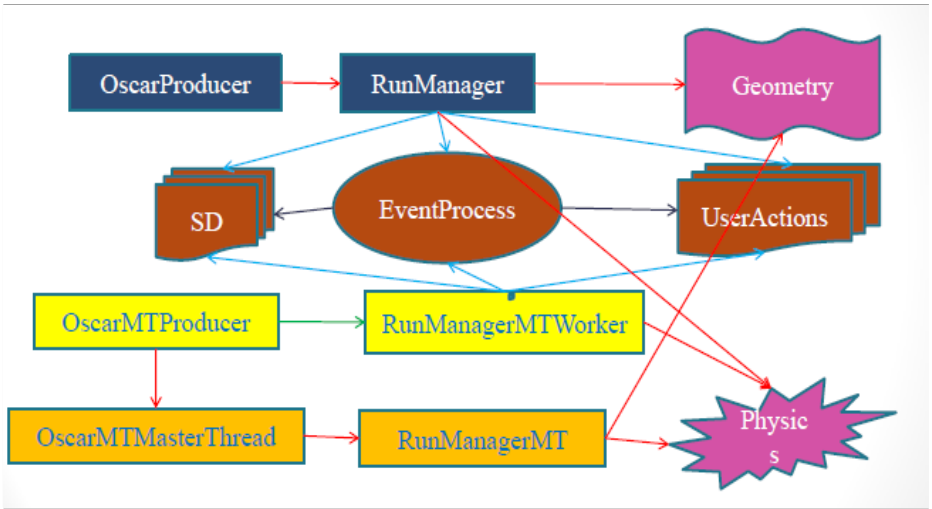

Figure 5. The scheme of the CMS Geant4 simulation. Blue boxes shows components working in the sequential mode, yellow boxes – in the MT mode.

After initialisation, the master thread is left alive for the duration of the job. Then, the module runs the worker-thread-specific Geant4 initialisation in its thread. Each time a module starts processing an event, it checks if the worker-thread-specific Geant4 initialisation has already been performed on that thread. If not, the module performs the initialisation. For these global and worker-thread-specific initialisations the threads are synchronized with mutexes and condition variables.

It is critically important, that geometry, magnetic field, physics, user actions, and sensitive detector classes are the same for both sequential and MT versions of the CMS simulation. This approach allows easy switch from one mode to another only modifying configuration scripts and perform detailed validation of simulation software. CPU and memory performances were studied using one of the standard CERN nodes with 12 cores, Geant4 10.0p03 compiled with gcc4.9.1 (figure 6). In order to use full CPU of the node in the sequential mode, 12 parallel runs have been executed to compare with one MT run with N threads. The MT run provides significant advantage in memory used (~3.5 MB for N=12) compared with the sequential run (~14 MB). Wall clock time for the same number of events are close to each other, so no CPU penalty is observed in the MT mode. In figure 6 time dependence on N is shown for different event type. There is no evidence of a speedup or a slow down if number of threads is above the number of physical cores.

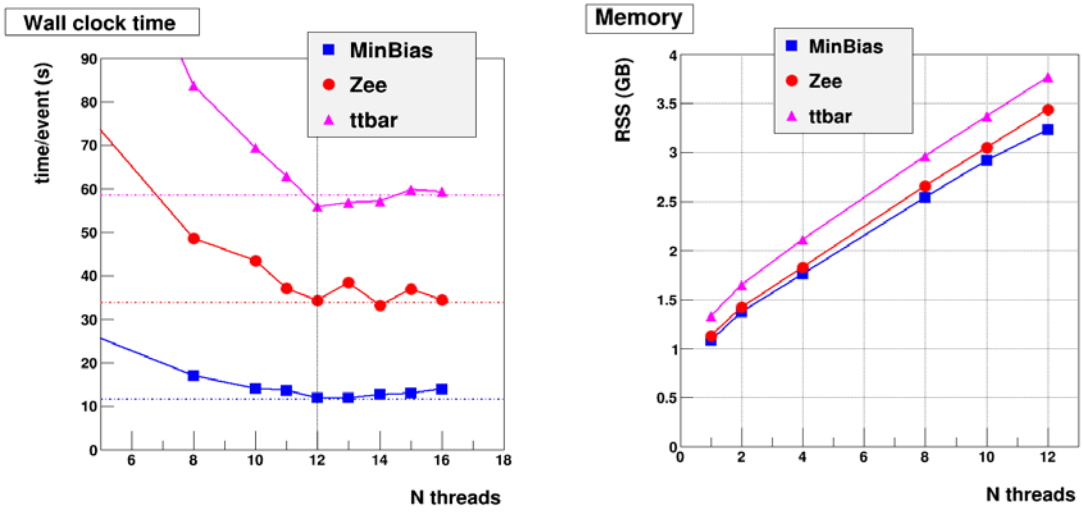

Figure 6. Wall clock time per event (left) and RSS memory (right) for different Monte Carlo event types as a function of number of threads. 12 core PC is used.

6. Summary

In view of the upcoming 13 TeV run several improvements have been introduced into simulation software of CMSSW. The Russian Roulette method has been developed and is used by default in Monte Carlo production. Together with other modifications it allows to speed up the CMS full simulation by approximately a factor of two. The new important capability, multi-threaded CMS simulation, is in preparation for a large scale production.

7. References

[1] The Geant4 Collaboration (Agostinelli S et al.) 2003 *Nucl. Instr. Meth. A* **506** 250-303
[2] Allison J et al. 2006 *IEEE Trans. Nucl. Sci.* **53** 270-78
[3] Banerjee S 2012 *J. Phys: Conf. Ser.* **396** 022003

[4] Abdullin S et al 2010 Calorimetry Task Force Report CMS-NOTE-2010-007; CERN-CMS-NOTE-2010-007 Geneva CERN 25 pp.
[5] Boudoul G et al. 2015 CMS Detector Description for Run-II and Beyond, these proceedings
[6] Lange D J 2015 Simulation and Reconstruction Upgrades for the CMS Experiment, these proceedings
[7] Hildreth M 2015 A New Pileup Mixing Framework for CMS, these proceedings
[8] Giammanco A 2014 *J. Phys: Conf. Ser.* **513** 022012
[9] Ivanchenko V N et al. 2014 *J. Phys: Conf. Ser.* **513** 022015
[10] Lewis E E and Miller Jr. W. F. 1984 *Computational Methods of Neutron Transport* (John Wiley & Sons, New York)
[11] Stephen A D and Stanley K F 2004 *A Monte Carlo Primer: A Practical Approach to Radiation Transport*, Volume **2** (Cluwer Academic/Plenum Publishers, New York)
[12] Cosmo G 2014 *J. Phys: Conf. Ser.* **513** 022005
[13] Jones C 2015 Using the CMS Threaded Framework in a Production Environment, these proceedings