CMS Simulation in the HL-LHC Era

Authors: S. Banerjee, D. Elvira, M. Hildreth, V. Ivantchenko, K. Pedro, I. Osborne, S. Sekmen, L. Sexton-Kennedy

Introduction

For its size and complexity, the data produced and handled by modern HEP experiments earned a place in the world of what is known as Big Data. For example, the CMS experiment produced, reconstructed, stored, transferred, and analyzed more than 10 billion simulated events during Run 1 in 2009-2013. The amount of data collected and stored by the LHC experiments by 2013 was on the order of 15 PB/year, not so far from the 180 PB/year uploaded to Facebook, the 98 PB of data in the Google search index, or the 15 PB/year in videos uploaded to YouTube[1]. By the end of 2016, the amount had increased to 50 PB/year. On the software side, simulation and reconstruction tools and algorithms are costly to develop and computationally expensive to run. Conservatively, the HEP field has spent of the order of 100 million US dollars just to develop and maintain simulation and reconstruction software for the large modern HEP experiments. In addition, in CMS, the simulation share of the cost of hardware lifecycle, maintenance, and operation is of the order of 5-10 million US dollars per year.

The CMS Collaboration has scheduled significant upgrades to the detector in order to meet the goals of the HL-LHC physics program. Furthermore, CMS expects to collect at least 150 times more data than in Run 1 through the end of the 2030s. The added complexity of the upgraded detector, the high luminosity environment, with up to 200 pileup interactions per event, and the need for much larger quantities of accurate full and fast simulation data will heavily tax the performance of both the simulation and reconstruction software, and pose severe stress on limited computing resources. It is therefore imperative to find solutions to speed up production and reconstruction of simulated events in order to cope with the increase in demand for computing resources in a time of flat budgets.

The HL-LHC CMS Detector

The CMS detector will be significantly upgraded for the HL-LHC run to start in 2026, including the vertex (pixel) detector, the tracker, the calorimeter, and the muon systems. The novelties are a high granularity end-cap calorimeter (HGC), which will use silicon or plastic scintillator as the active material in different compartments, and a fast timing system using silicon or LYSO crystal scintillator. The CMS simulation group has developed a Geant4-based simulation package for this “phase 2” detector with the goal to aid the experiment in the design optimization and characterization of the new detector components. CMS is actively working on the following tasks to address the HL-LHC era challenges:

- Tools for detector description. Targeting Run 1, CMS developed a detector description language for writing XML instance documents capable of describing the CMS detector for offline simulation and reconstruction software within the CMS software applications. The challenge for phase 2 is to improve the algorithmic descriptions of complex detector shapes without loss of computing performance. Alternatives to XML are also being explored. The development of a common language for detector description across experiments offers an interesting opportunity for collaboration in the interest of reducing effort duplication and saving resources.

- Geant4 physics validation monitoring and benchmarking using test beam experiments. Simulation software in CMS is continuously updated with new versions of Geant4. The evolution of electromagnetic and hadronic physics models is monitored and validated using test beam data collected by CMS using prototypes of the HGC calorimeter modules in a controlled environment. A software tool has also been developed to validate Geant4 physics models with real collider data. It makes use of isolated charged hadrons measured simultaneously in the tracking detector as well as the calorimeter.

- Phase 2 detector simulation computing performance monitoring. Simulation, including production and analysis, from startup to May 2016, took 85% of the computing resources available to the experiment. The Geant4 part alone consumed 40% of the total. Preliminary measurements based on a standalone version of the Geant4 CMS simulation application indicate that it will take between 25% and 70% more CPU time per event to run the CMS simulation for the phase 2 detector, depending on the selected set of physics models (physics lists). These studies will be followed by code review and optimization based on profiling studies and regular computing performance monitoring.

Pileup Interactions

Another aspect of simulation that poses significant challenges is the modeling of pileup interactions. Here, the issues are related to I/O and local network bandwidth at production sites. To simulate pileup interactions for the luminosities that will be encountered at the HL-LHC, one must merge the particle contents of 200 peripheral interactions for each bunch crossing that occurs when the detector elements are sensitive. This often requires the consideration of many collisions before and after the central hard-scatter event that is the focus of the generation. Thus, thousands of individual minimum bias interactions may eventually be required to simulate the pileup for one hard-scatter event. This puts huge strains on the local networks at production sites as these events are served out to the compute nodes. CMS has developed a strategy of “pre-mixing” the peripheral interactions in a manner such that the hits can be merged with those from a hard-scatter event in a later production step. The pre-mixed events are produced at a site with high local network bandwidth and then distributed for the global production of Monte Carlo samples. The pre-mixed pileup events are larger than a “typical” hard-scatter event, but not dramatically so. The advantage is that only one pileup event is now required per hard-scatter event, which dramatically decreases the network load.

A potential evolution to the pre-mixing approach is the “data mixer”, a tool that would allow CMS to overlay real zero-bias data onto a MC hard scatter event to model the effects of detector noise, pileup and any other beam-related signals induced in the detector. The Data Mixer may also be used to overlay MC-on-MC, as well as data-on-data events. Mixing of collider zero-bias data onto MC hard scattering events would replace the current pre-mixing procedure, which is based on the overlay of simulated min-bias events to model detector and pileup effects. Pre-mixing utilizes the machinery developed with data mixing in mind, which performs the overlay at the raw data level. Valid data mixing will require a thorough understanding of possible non-linear effects from photo-detectors and readout electronics, as well as descriptions of the detector geometries for simulation and real data that are compatible (i.e., the sensors are in the same locations in both geometries within some tolerance). The effects of zero-suppression, which will necessarily be applied in detector readout on low energy hits, and how well these are represented in the data-data combination procedure will also need to be studied.

Computing Performance

Estimates are that CMS computing needs will increase by a factor of 10 to 100 in the High-Luminosity LHC (HL-LHC) era, depending on the solutions developed to face simulation, pileup, and reconstruction challenges. On the simulation side, the Geant4 Collaboration has dedicated significant efforts to improve the toolkit computing performance during the last few years, as code was reviewed and optimized. The introduction in Geant4 of event-level multithreading capabilities in 2013 brought significant memory savings for CMS. While time performance did not improve with multithreading, deviating from perfect scaling by approximately 10% when executing on 30 cores, memory use for the CMS simulation application had been significantly reduced with 170 MB used for the first event, and only 30 MB per event for each additional thread. Through code optimization and improvements to the Geant4 engine and physics algorithms, the average time performance improvement through the life of the LHC experiments (2010-2015) has been of the order of 35%. Remarkably, the percentage time performance improvement during this period is in the double digits even as the physics models were improved significantly for accuracy, something that comes typically associated with a time performance penalty.

Although the Geant4 team strives for improving the toolkit computing performance further, the code is already highly optimized, with no hot spots of computing performance, a fact that makes it difficult to achieve additional large factors in time performance in the future. On the hardware side, transistor density growth is not enough to keep up with the expected increase in computing needs described above. Although this growth is more or less keeping up with Moore’s law, doubling every couple of years, clock speed has been flat since approximately 2003. Therefore, solutions must be found elsewhere, leveraging the core count growth in multicore machines, using new generation coprocessors, and re-engineering code along the lines of the new programming paradigm based on concurrency and parallel programming. For example, a hybrid-computing model based on coprocessors or accelerators would allow sharing of work across a mixture of computers with different architectures. Each processor type could be used to perform different tasks depending on the nature of the task. With the goal to achieve significant time performance improvements in simulation code, expert teams are invested in R&D programs to explore the potential of multi threaded track-level (particle-level) parallelization, improved instruction pipelining, vectorization and single instruction multiple data (SIMD) architectures, and data locality. For example, the GeantV project to develop a next generation detector simulation toolkit has set a goal achieve a speedup factor of 2 to 5 while enhancing physics accuracy at the same time.

Improved versions of Geant4 with better physics and computing performance, vectorized geometry and physics libraries optimized for SIMD coming from the GeantV project, a prototype of the GeantV engine designed from the ground up for parallel programing and modern computing architectures, HEPCloud resources, including HPC facilities hosted by research institutions and universities around the world, are all tools that CMS should be ready to test and eventually use in order to address the HL-LHC challenges. Examples of planned activities are:

- Testing, integration, and computing performance evaluation of new Geant4 releases as they become available.

- In the context of the Geant4-based CMS application, testing, validation, and eventual integration of the VecGeom geometry library, developed for optimal performance with GeantV but expected to provide computing performance gains also with Geant4.

- At a later stage, and also in the context of the Geant4-based CMS application, testing, validation, and eventual integration of vectorized and improved physics libraries developed within the GeantV project, such as for example an EM physics library with improved physics.

- Assuming that GeantV delivers a full prototype of its engine and libraries by 2018, and that the speedup factor is significant, as promised, CMS will test the new tool and evaluate potential migration on the timescale of the HL-LHC program, scheduled to start in 2026. The first step, ongoing, consists of integration tests of the current version of the GeantV engine within a toy version of the CMS software framework. The goal of the latter is to verify consistency between the multithreading approaches.

Fast Simulation

CMS has developed a fast detector simulation package (FastSim), which serves as a fast and reliable alternative to the detailed Geant4-based simulation (FullSim), and enables efficient simulation of large numbers of standard model and new physics events. FullSim is dominantly used for standard model processes, which need to be accurately modeled as signals in measurement studies or backgrounds in new physics searches. While the order of magnitude of the CPU time per FullSim event is minutes, FastSim reduces simulation time by a factor of ~100 and simulation plus reconstruction time by a factor of ~20 by using a simplified geometry with infinitely thin material layers and simple analytical material interaction models that are parametrized and tuned to agree with FullSim. Therefore, the CMS FastSim tool provides a much faster alternative that reproduces FullSim with a ~10% accuracy for kinematic distributions such as the momentum and pseudorapidities of high level event objects such as light jets, b jets, missing transverse energy, muons, electrons, photons and taus. Higher discrepancies are observed when comparing variables that depend on structure or shape properties such as jet particle composition or variables used for electron and photon identification.

An effort to refactor the FastSim code has recently started to make FastSim more configurable and automate the tuning procedures in preparation for the large demand for fast and accurate simulation in the HL-LHC era. This work includes to:

- Implementation of the upgraded detectors in the current framework in a configurable and flexible way.

- Explore GeantV as an alternative framework for FastSim. GeantV plans to embed fast simulation in its framework, providing generic tools to correlate output information, and a library of fast simulation algorithms for trackers and calorimeters. A GeantV-based FastSim application would provide common, concurrent, infrastructure for scheduling work and dataflow that will build libraries of parametrized quantities derived from detailed simulation, and will provide tools for FastSim-to-FullSim tuning and comparison.

- Explore the possibility of an ultra fast self-tuning non-parametric simulation. A re-emerging alternative trend in simulation is to develop ultra fast, self-tuning simulators based on lookup tables that directly map generator events into simulated events. A past example was Turbosim developed at the Tevatron for the D0 and CDF experiments. A redesign called Falcon was recently released, with a more efficient algorithm for the multi-dimensional mapping of particle properties.

Summary and Outlook

CMS has identified the software, computing, and physics challenges associated with the simulation needs of the HL-LHC program. They mostly arise from the increased computing cost of simulating big data for the more complex upgraded detectors, with larger pileup contributions coming from the high luminosity environment, in a time of limited resources. New programming paradigms based on fine-grained parallelization and modern computing architectures offer opportunities to develop solutions that would close the gap between the needs and the available resources.

[1] https://www.wired.com/2013/04/bigdata