Optimizing the ATLAS Geant4 detector simulation software

Evangelos Kourlitis[1], Akanksha Vishwakarma[2], Andrei Sukharev[3], Benjamin Michael Wynne[2], Benjamin Morgan[4], Caterina Marcon[5], Dongwon Kim[6], Evgueni Tcherniaev[7], Guilherme Amadio[7], John Apostolakis[7], John Derek Chapman[8], Marilena Bandieramonte[9], Miha Muskinja[10], Mihaly Novak[7], Mustafa Andre Schmidt[11], Tommaso Lari[5] and Walter Hopkins[1]

[1] Argonne National Laboratory (US), [2] The University of Edinburgh (GB), [3] Budker Institute of Nuclear Physics (RU), [4] University of Warwick (GB), [5] INFN Sezione di Milano (IT), [6] Stockholm University (SE), [7] CERN (CH), [8] University of Cambridge (GB), [9] University of Pittsburgh (US), [10] Lawrence Berkeley National Laboratory (US), [11] Bergische Universitaet Wuppertal (DE)

E-mail: evangelos.kourlitis@cern.ch

Abstract. The ATLAS experiment at the LHC relies critically on simulated event samples produced by the full Geant4 detector simulation software (FullSim). FullSim was the major CPU consumer during 2018, the last year of Run 2 data-taking, and it is still expected to be significant in the HL-LHC era. In September 2020 ATLAS formed the Geant4 Optimization Task Force to optimize the computational performance of FullSim for the Run 3 Monte Carlo campaign. This report summarizes the already implemented and upcoming improvements. These include improved features from the core Geant4 software, optimal options in the simulation configuration, simplifications in geometry and magnetic field description and technical improvements in the way ATLAS simulation code interfaces with Geant4. Overall, more than 50% higher throughput is achieved, compared to the baseline simulation configuration used during Run 2.

ATL-SOFT-PROC-2023-002
13/03/2023

1. Introduction

Simulated event samples play a crucial role in the design, calibration and result interpretation of High Energy Physics (HEP) experiments, such as the ATLAS experiment at CERN [1]. During the LHC Run 2 period, which concluded in 2018, the simulation of the passage of particles through the detector medium using solely the Geant4 toolkit [2] (FullSim) was the workflow consuming the most computing resources. This trend is expected to continue in both the Run 3 and HL-LHC periods. Especially for the HL-LHC, the situation deteriorates as computing budget calculations highlight the stress between the required resources and the projected available ones [3].

To tackle this challenge, the ATLAS Geant4 Optimization Task Force was formed in September 2020 with the purpose to optimize the computational performance of FullSim for the upcoming Run 3 Monte Carlo campaign. The mandate targeted at least a 30% CPU time speedup with respect to the Run 2 software. This report provides an overview of the

improvements that have already been implemented into the production workflow and summarizes some of those currently under active development and validation. Those include 1) new features from the core GEANT4 software and technical improvements in the way ATLAS simulation code interfaces with GEANT4, 2) simplifications in the simulation of physics processes and the geometry and magnetic field description, and 3) optimizations in the simulation configuration, which are specific to the ATLAS detector.

The document summarizes the improvements already implemented and three upcoming ones, in Sections 2 and 3 respectively. Finally, measurements of the resulting software performance are presented in Section 4. Conclusions and prospects are given in Section 5.

2. Improvements in Production

2.1. GEANT4 *Static Linking*

This improvement concerns the way the ATLAS software – referred to as *Athena* henceforth – is built and interfaces with GEANT4. In particular, it focuses on the linking of GEANT4, and different link types were evaluated for their performance. The nominal configuration for Run 2 was dynamic multi-library linking while dynamic single library and static linking were tested for Run 3. The dynamic single library showed an increase in execution time, which was attributed to the trampoline/lookup table mechanism of dynamic linking. This leads to increased calls and jumps, slowing down the simulation execution. Static linking was found to be the best performer and to enable it in Athena, all packages linking to GEANT4 had to be consolidated into a single *Big Library* that could then be statically linked to Athena. Comparison of execution times for a benchmark application for the different linking types is illustrated in Figure 1, where the static linking showed an improvement of up to 7%.

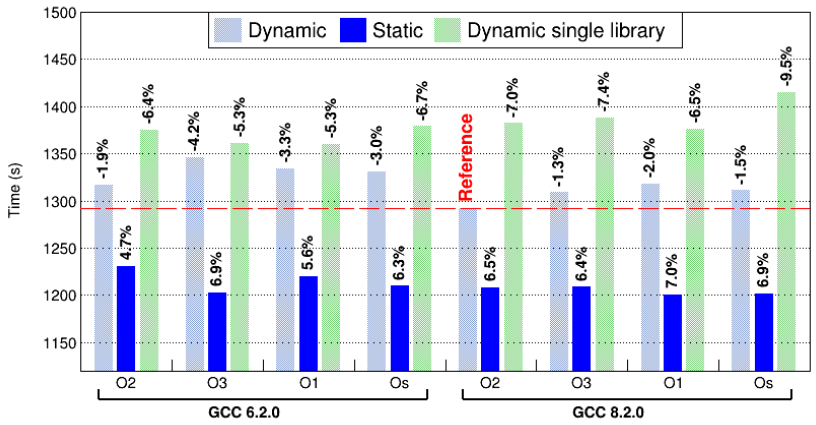

Figure 1. Comparison of execution times for a standalone GEANT4 simulation of the ATLAS detector, excluding the electromagnetic end-cap calorimeter. Differences in performance are expressed as a relative percentage with respect to the reference case of the Run 2 simulation configuration. Each of the benchmark run five times and the average value is quoted, standard deviations are of the order of 2%.

2.2. General Gamma Process

In segmented detectors with different materials GEANT4 particle propagation is interrupted by the geometrical boundaries. In those boundary crossings GEANT4 calculates all the cross-sections of the various physics processes a particle can undergo in the new material. Particularly for photons, a new *general* process has been introduced that sums up all the possible physics

process, thus only a single cross-section evaluation occurs at the geometrical boundaries. This `G4GammaGeneralProcess` provides a single access point to the process manager, therefore the number of instructions is reduced. Its usage in ATLAS simulation has been tested on 100 top-pair events and a CPU speedup of about 4.5% was measured.

2.3. Physics Processes Simulation Simplifications

2.3.1. Electromagnetic Range Cuts For some electromagnetic physics processes the cross-section at low energies is extremely high and the simulation consequently expensive. To address this, a production threshold is employed by GEANT4 to exclude all particles below a certain energy from being generated. This is referred to as a *range cut* and is provided by the user in units of length, which is then internally translated to energy, given the material and particle type. The secondary particles are then not produced and their energy is deposited across the simulation step length of the primary particle. In the ATLAS GEANT4 simulation range cuts are enabled for physics processes producing electrons, namely Compton scattering, photon conversion and the photoelectric effect, which resulted in 6-7% CPU speedup. The reduction in the number of electrons in the simulation can be seen in Figure 2.

2.3.2. Neutron and Photon Russian Roulettes In the simulation of ATLAS calorimeters, which are the most resource-intensive components of the whole detector simulation, a large portion of the CPU time is consumed by neutrons and photons. The *Russian Roulette* technique is used to reduce this time by randomly discarding neutrons and photons below a certain energy threshold and weighting the energy deposits of the remaining particles accordingly. The effect of different thresholds on neutron tracks can be seen in Figure 3. For the Run 3 campaign, a 2 MeV threshold was chosen for neutrons and 0.5 MeV for photons produced in the LAr calorimeter. This configuration resulted in a 10% improvement in simulation CPU time.

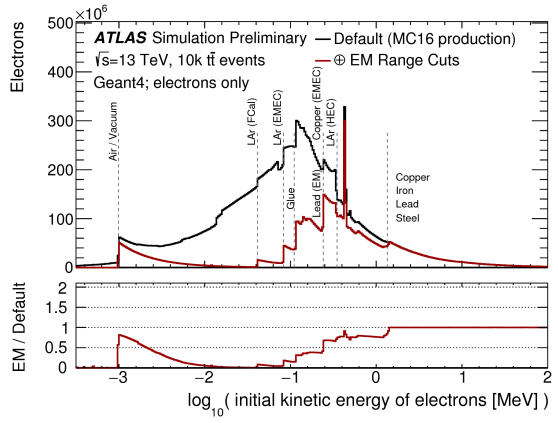

Figure 2. Distribution of the kinetic energy of electrons in the ATLAS GEANT4 simulation. The black curve shows the distribution for the nominal setup and the red curve shows the distribution after enabling range cuts for electromagnetic processes. Vertical lines indicate range cut values for some components/materials.

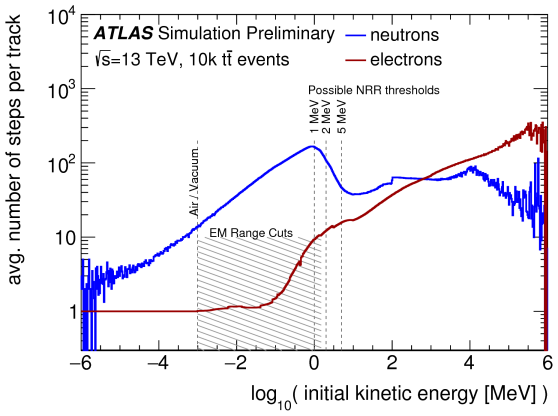

Figure 3. Average number of simulation steps per track as function of the particle kinetic energy in the nominal setup. Vertical lines indicate the potential energy threshold for the Neutron Russian Roulette (NRR) algorithm and the hatched box indicates where range cuts have the largest effect.

2.4. Simplified Geometry and Magnetic Field Description

2.4.1. EMEC Shape Variants For the Run 3 campaign the description of the ElectroMagnetic EndCap (EMEC) calorimeter [4] has been improved and two new variants were introduced in addition to the nominal one used in Run 2. The first new variant is referred to as *Cones* and it reduces the usage of `G4Polycones` by employing an improved shape, the `G4ShiftedCone`. In this configuration, the outer wheel is divided into two cone-shaped sections. The second new variant is referred to as *Slices* and it reduces the time required for geometry navigation calls by dividing each wheel into many thick slices along the Z axis. Benchmarking showed that the Slices variant was the most efficient, resulting in a CPU speedup of 5-6% with respect to the nominal configuration.

2.4.2. Magnetic Field Tailored Switch-off This optimization concept is based on the observation that it is possible to switch the simulation of the magnetic field off, in a region deep inside the barrel section of the Liquid Argon calorimeter [4], without significant impact on the modeling of the shower shape observables. This switch-off mechanism is applied to the simulation of all the particles, except muons, and is used in the Run 3 simulation campaign. Benchmarking with 200 top-pair events showed a 3% CPU speedup, while this approach has potential applications in other regions of the detector as well where the magnetic field is minimal.

3. Improvements in Development and Validation

3.1. VecGeom Usage

VecGeom is a geometry modeller library with hit-detection capabilities, optimized for both vectorized (SIMD) and scalar data inputs [5]. It boasts efficient geometry primitives and navigation algorithms, especially for complex geometric shapes. For the Run 3 Monte Carlo campaign VecGeom has been enabled in the ATLAS GEANT4 simulation for specific shapes that were shown to improve computational performance (cones and polycones). The use of VecGeom for these shapes resulted in a CPU speedup ranging from 2-7% depending on the compute platform.

3.2. Voxel density tuning

The size of the voxels used internally by GEANT4 to optimize the geometry description and navigation can be adjusted through a density parameter. The purpose of this project is to find the optimal values of this parameter that balance the simulation CPU time and memory usage for the detector description. The simulation accuracy should not significantly be affected by this parameter. Figure 4 shows the average simulation time per event and the amount of memory used for detector description, per sub-detector, as a function of the voxel density. Smaller voxel density values result in lower memory usage but longer simulation times. The standard value in GEANT4 is 2. The current memory consumption for the description of the ATLAS detector, including an optimized value of 0.1 for the Muon system and 0.5 for a part of the LAr calorimeter, is 111 MB. By applying improved values from this analysis to the SCT, TRT, Tile, and Pixel sub-detectors, the memory consumption could be reduced to 65 MB, resulting in a 40% improvement. Further optimization of the full LAr calorimeter could bring the memory consumption down to 56 MB, i.e. a 49% improvement.

3.3. Woodcock Tracking

Woodcock Tracking is a tracking optimization technique suited to highly segmented detectors where the simulation steps are limited by the geometric boundaries rather than physics interactions [6]. It operates by performing tracking in a unified geometry made by the densest

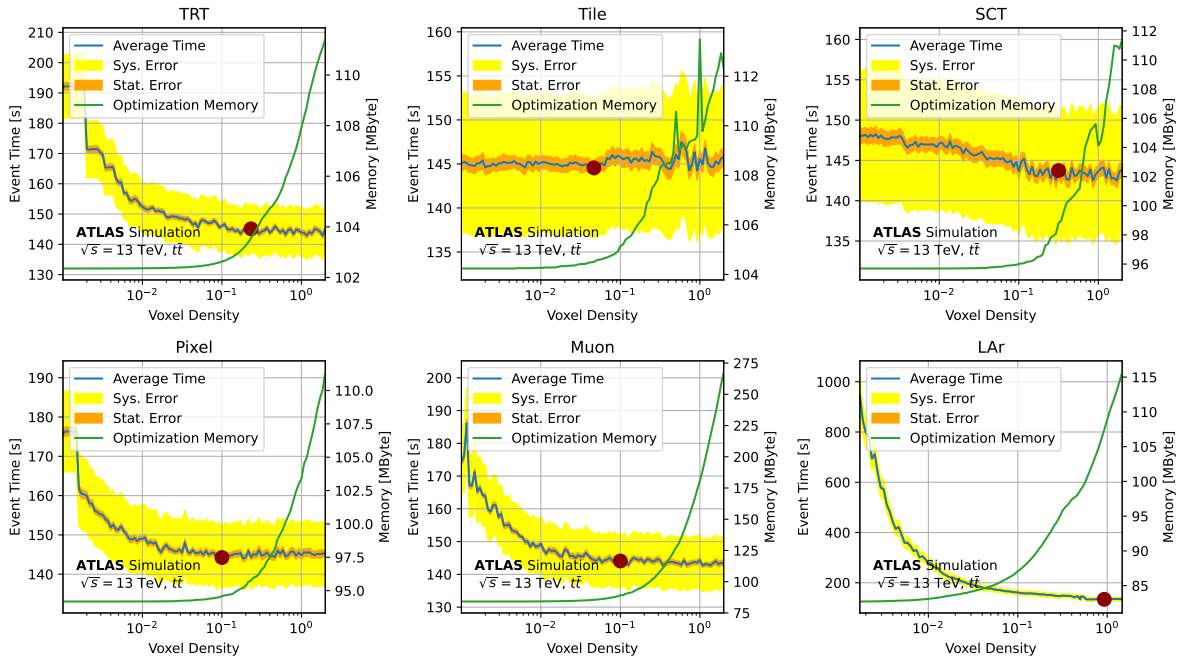

Figure 4. The average simulation time per event and memory used for the geometry description as a function of the voxel density value used for six ATLAS sub-detectors. The yellow band indicates the calculated uncertainty from simulating 50 top-pair events, while the orange band shows the resulting statistical uncertainty from simulating each event 10 times. The red marks indicate the chosen voxel density size for each sub-detector.

material. An additional fictitious δ-interaction is introduced with macroscopic cross-section:

$$\Sigma_\delta(E, material) = const. - \Sigma_\gamma(E, material), \tag{1}$$

such that the total macroscopic cross-section of the geometry remains constant:

$$\Sigma(E) = \Sigma_\delta(E, material) + \Sigma_\gamma(E, material) = const. \tag{2}$$

Using $\Sigma(E)$ to sample the step length until the next (real or fictitious) interaction eliminates the need for stopping at geometric boundaries. Then the probability of a real photon interaction is calculated as:

$$P_\gamma(E, material) = \Sigma_\gamma(E, material)/\Sigma(E). \tag{3}$$

In this way, fewer simulation steps and cross-section evaluations occur while preserving the physics results. The Woodcock Tracking technique has been developed and tested for the EMEC sub-detector and preliminary results showed a simulation time speedup of 8-9%.

4. Performance Studies

In this section measurements of the performance of the software before, during and after the implementation of the various aforementioned improvements are presented. For benchmarking the software a particular test machine was used, employing an AMD EPYC™ 7302 processor clocked at 3 GHz. For those tests the CPU boosting and simultaneous multithreading capabilities of the processor were disabled. In Figure 5 the evolution of the simulation time per event is shown after employing each of the improvements described. Overall, and with respect to

the legacy Run 2 software, 33% CPU speedup is achieved, which corresponds to 50% higher simulation throughput[1]. It should be noted that the various optimizations are correlated, thus the improvement does not add up linearly.

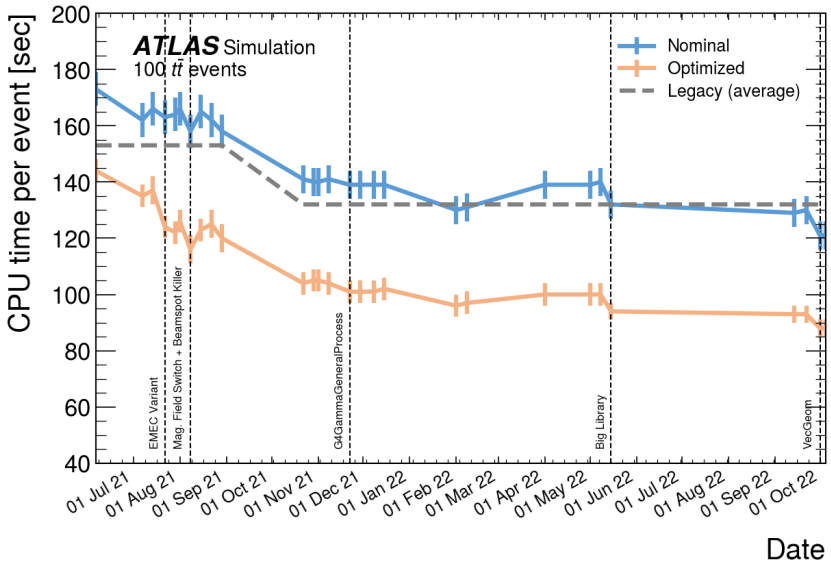

Figure 5. Simulation CPU time per event evolution from July 2021 until October 2022. The blue line is measured from the nominal Run 3 software while the orange after applying the aforementioned optimizations. The Big Library and VecGeom optimizations affect both the nominal and the optimized software versions. The grey line indicates an average time measured from the legacy software used in Run 2. In October 2021 the new benchmark machine mentioned in the text was introduced thus the scale of all the measurements changed.

Beyond this benchmarking, which ran on a local machine, the software performance was also measured in a realistic production environment at a Worldwide LHC Computing Grid (WLCG) site. The results can be seen in Figure 6 where the performance improvement can be verified on the simulation of 100,000 top-pair events. From the mean values of the simulation CPU time it can be seen that a 36% faster software has been delivered.

5. Conclusions

In order to increase the computational efficiency of the ATLAS FullSim, the ATLAS Geant4 Optimization Task Force was formed and over the past two years several improvements have been developed and deployed for the Run 3 Monte Carlo campaign. These include advancements in the core Geant4 software, optimizations in the interface between the ATLAS simulation code and Geant4, simplifications in the simulation of physics processes and geometry and magnetic field description, and specific tunings of the simulation parameters. Overall, the optimized software delivered for production can simulate 50% more events while utilizing the same computational resources as the Run 2 software.

The effort to improve the ATLAS FullSim continues with additional optimizations, such as the voxel density tuning and Woodcock tracking technique. Further improvements in

[1] The *throughput* is defined as the number of simulated events in the unit of time.

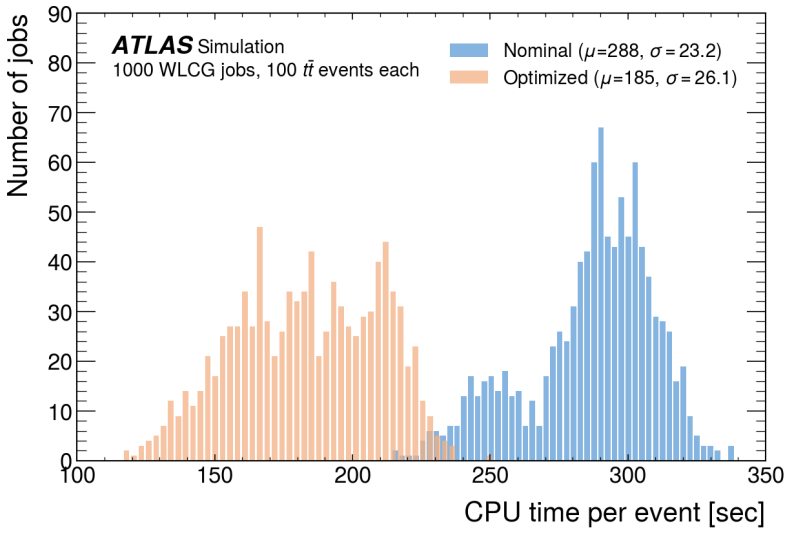

Figure 6. Distributions of ATLAS GEANT4 detector simulation CPU time per event using the legacy software used in Run 2 (Nominal) and the optimized software for the Run 3 simulation campaign (Optimized). The Big Library optimization has been excluded from these tests. The benchmarks comprise 1000 jobs simulating 100 top-pair events each at the Brookhaven National Lab WLCG Tier 1 cluster. The mean value (μ) and standard deviation (σ) of the distributions are also indicated.

the description of EMEC will allow faster navigation and provide data structures that allow simulation on accelerator hardware. Additionally, tuning of the simulation parameters governing the propagation of particles in the ATLAS magnetic field as well as avoid simulating particles produced around the LHC beam pipe that never leave a signature in the detector will speed up the simulation further. Finally, advanced compiler optimizations are also pursued. Those developments, among others, will be incorporated into the software over the course of Run 3 and will lead to a highly tuned and efficient workflow to meet the computational budgets of the HL-LHC.

References

[1] ATLAS Collaboration 2008 *JINST* **3** S08003
[2] Agostinelli S *et al.* (GEANT4) 2003 *Nucl. Instrum. Meth. A* **506** 250–303
[3] Calafiura P, Catmore J, Costanzo D and Di Girolamo A 2020 ATLAS HL-LHC Computing Conceptual Design Report Tech. rep. CERN Geneva URL `https://cds.cern.ch/record/2729668`
[4] ATLAS Collaboration 1996 ATLAS liquid-argon calorimeter: Technical Design Report Tech. rep. CERN Geneva URL `https://cds.cern.ch/record/331061`
[5] Wenzel S, Apostolakis J and Cosmo G 2020 *EPJ Web Conf.* **245** 02024
[6] Woodcock E 1965 *Proceedings of the conference on applications of computing methods to reactor problems*