PROCEEDINGS OF SCIENCE

The ATLAS Monte Carlo detector simulation for Run 3 at the LHC

John Derek Chapman[a,*] **on behalf of the ATLAS collaboration**

[a] *University of Cambridge, Cambridge, UK*

E-mail: chapman@hep.phy.cam.ac.uk

In preparation for Run 3 at the Large Hadron Collider, the Monte Carlo Detector Simulation performed with Geant4 (Full Simulation) within the ATLAS experiment has undergone significant improvements to enhance its computational performance and overall efficiency. These proceedings offer a comprehensive overview of the optimisations implemented in the ATLAS Full Simulation for Run 3. Notable developments include the application of EM range cuts, the implementation of Neutron and Photon Russian roulette and the recent development of the Woodcock tracking in the Electromagnetic Endcap Calorimeter, a critical subdetector that for its peculiar structure represents a significant consumer of CPU resources in the ATLAS simulation. Beyond optimisations targeting specific detector components, the improvements include the tuning of simulation parameters (including the magnetic field), smarter and more efficient geometry descriptions, the implementation of new Geant4 core features and improvements that target the way Geant4 is linked and used within the ATLAS simulation framework. These enhancements collectively resulted in a remarkable achievement, with a speed-up in CPU time of a factor of 2 compared to the baseline simulation configuration used in Run 2. In addition to showcasing the Run 3 ATLAS simulation, this contribution provides an overview of forthcoming optimisations, emphasising both immediate and longer-term enhancements.

ATL-SOFT-PROC-2024-010
11 October 2024

42nd International Conference on High Energy Physics (ICHEP2024)
18-24 July 2024
Prague, Czech Republic

[*] Speaker

1. Introduction

The ATLAS software underwent a major programme of development between Run 2 and Run 3 [1] [2]. The main focus of simulation development was on improving the throughput of the simulation. The ATLAS Monte Carlo (MC) simulation for Run 3 of the Large Hadron Collider (LHC) is produced using version 10.6.patch03 of the GEANT4 tool kit [3] within the Athena framework [4].

2. Competing factors for simulation software development

Three competing factors must be considered when developing simulation software:

1. Stability of physics modelling for analysers

2. Improved physics modelling (data-MC agreement)

3. Improved technical performance (higher throughput, lower resource requirements)

ATLAS typically analyses the data of each LHC run separately. Analysers therefore require a set of MC samples with consistent physics modelling corresponding to all data periods in an LHC run with which to perform their analysis. This set of samples is known as an MC campaign. Physics modelling changes in MC require updated recommendations for physics objects. This is an expensive process (in terms of personpower) and so is only undertaken once per MC campaign. If the physics modelling were to be updated then most analysis work would also need to be checked. For these reasons, the convention in ATLAS is not to alter the physics modelling in simulation during an LHC run unless a bug is found which makes the samples unusable.

Separate MC samples are produced for each data-taking year in an LHC run to reflect changes in beam properties (these are known as sub-campaigns). Typically a new production software release is used for each sub-campaign. It is desirable to produce MC samples as efficiently as possible. Potential optimisations are carefully validated to determine whether or not they alter the physics modelling in a way that would be perceptible to analyses. Optimisations which improve technical performance, but do not alter the physics modelling, can be added to production releases between sub-campaigns. Over time ATLAS aims to improve the physics modelling (data-MC agreement); this is done in close collaboration with GEANT4. However, changes (improvements) to physics modelling can only be included in production releases at points when new physics object recommendations are planned.

Finally, there is a tension between improving technical performance (for example with fast simulation techniques) and improving physics modelling (for example with more detailed models).

3. Methods of optimising technical performance

There are two main ways to optimise the technical performance of simulation code: either by avoiding simulating uninteresting particles (output changes) or by speeding up the simulation of interesting particles (output may change). Optimisations in the second category can be further divided up into those which do the same thing, but faster (output unchanged) or those which do something simpler (output changes).

As mentioned in the previous section, output-changing optimisations require very careful validation, but can produce output which is compatible with being analysed together with previous MC production. The following sections describe the optimisations implemented for the Run 3 MC campaign.

3.1 Avoid simulating uninteresting particles

A number of optimisations in this category have been applied for the Run 3 MC campaign:

Beam-pipe killer - Particles entering certain forward beam-pipe volumes will be very unlikely to reach sensitive detector regions. The "beam-pipe killer" is a Geant4 Fast Simulation model which kills particles entering these volumes. This will not directly affect the physics output, but statistical differences in output are expected due to changes in the random numbers used in the simulation of remaining particles.

Neutron/Photon Russian Roulette - Kill low energy neutrons/photons with some probability when created. Upweight energy deposits of surviving particles accordingly. The measured stand-alone speed-up is $\sim 10\%$

EM Range Cuts - Range cuts are a built-in way of optimising Geant4 performance. For each material-volume pair, range cuts can be specified in distance units (mm). Secondary particles that are expected to travel less than the range cut are not created and their energy is immediately deposited by the parent particle. By default Geant4 does not apply range cuts for the conversion, photo-electric or Compton-scattering gamma processes.[1] Activating EM range cuts greatly reduces the number of particles created by Geant4, as can be seen from Figure 1. A range cut of 0.1 mm is used (matching what is used for electron processes). Measured stand-alone speed-up of $6 - 10\%$.

3.2 Simulate interesting particles more efficiently

A number of optimisations in this category have been applied for the Run 3 MC campaign:

VecGeom [6] - ATLAS uses VecGeom implementations for a subset of `G4Solid` implementations. In testing it was found that switching only Cones, Tubes and Polycone implementations from the defaults to VecGeom gave the best performance. The measured stand-alone speed-up is $2 - 7\%$

EM Endcap Calorimeter (EMEC) Geometry Optimisation - Reduce the time needed for geometry navigation calls by dividing custom solid for the EMEC inner (outer) wheel volume into 14 (21) thick slices along the z-axis. The measured stand-alone speed-up is $5 - 6\%$

Tailored magnetic field switch-off in LAr Calorimeters - Magnetic field switched off in central LAr calorimeter for all particles except muons. The measured stand-alone speed-up is 3%.

`G4GammaGeneralProcess` - Use a super-process that hides all six standard physics processes involving photons. Then only one mean free path needs to be calculated for a photon. The measured stand-alone speed-up is 3%

Woodcock Tracking in the EMEC - Woodcock Tracking [7] is a tracking optimisation technique for highly segmented detectors where the geometry boundaries rather than the physical interactions

[1] An option is provided by Geant4 to activate range cuts for these processes: `'/process/em/applyCuts true'`.

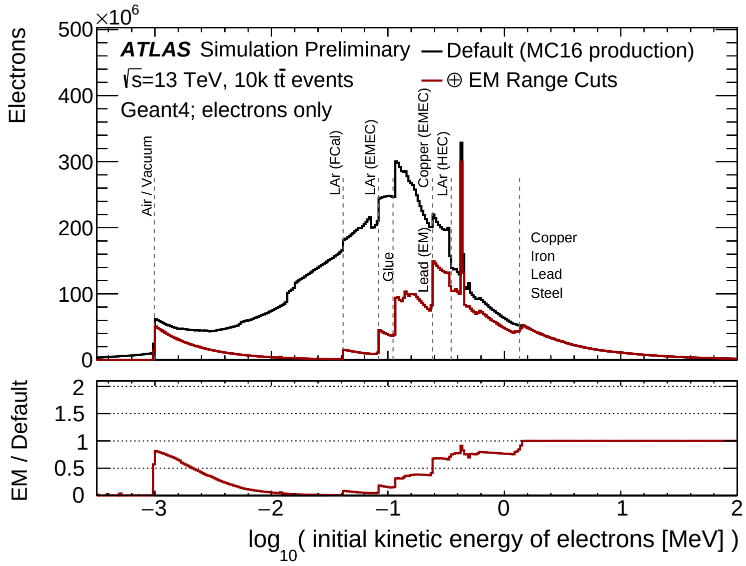

Figure 1: Distribution of the initial kinetic energy of electrons in the ATLAS GEANT4 simulation. The black curve shows the distribution for the default setup (MC16 production) and the red curve shows the distribution for the default setup plus the added range cuts for electromagnetic GEANT4 processes ('conv', 'phot', 'compt'). Vertical gray dashed lines indicate range cut values for some materials and the right-most dashed line indicates an area with multiple range cuts in close proximity for various metals. [5]

limit the simulation steps. In the EMEC region, photon simulation is dominated by the transportation process. Photons don't interact during transport (no continuous energy deposition), therefore it is safe to perform tracking of photons in a simplified EMEC geometry (i.e. without boundaries) made of the densest material from the standard EMEC geometry (Pb). Interactions then occur with a probability equal to ratio of cross-sections of the true material and Pb. A GEANT4-based implemenatation of Woodcock tracking [8] is applied on top of the `G4GammaGeneralProcess`. This code will be part of `G4HepEM` [9], but was added as a patch to the ATLAS version of GEANT4 10.6.patch03.[2] This gives a measured stand-alone speed-up of 17.5% due to a 50% reduction in number of steps simulation for photons in the EMEC. Only statistical changes in output are observed.

Big Library - Athena builds typically create a shared object library (`.so`) per package. The program then determines at runtime which `.so` files to load into memory, saving memory compared to the case of a single monolithic executable. However look-ups of function calls between different `.so` files is slow. As most `.so` files which link to GEANT4 are used in most simulation jobs there is potential for optimisation by building a single shared object library for all Athena code with GEANT4 dependencies - statically linked to GEANT4 libraries. The measured stand-alone speed-up is 5 – 7%.

Link Time optimisation (LTO) - Once all files have been compiled separately into object files, traditionally, a compiler links (merges) the object files into a single file, the executable. However in LTO the compiler is able to dump its intermediate representation, so that all the different compi-

[2]GEANT4 command: `'/process/em/useWoodcockTracking EMEC'`.

lation units that will go to make up a single executable can be optimised as a single module when the link finally happens. As all Athena code with GEANT4 dependencies is statically linked together into a single shared-object library it is possible to use LTO on this shared-object library (instead of an executable). This only required changes in the CMake configuration of the Athena build. As expected, the simulation output is identical after this purely technical change. The measured stand-alone speed-up is $\sim 5\%$

3.3 CPU performance

As Figure 2 shows, throughput increased by a factor of 1.84 between the Run 2 and Run 3 MC campaigns despite an increase in $\sqrt{s}$ from 13.0 TeV to 13.6 TeV.

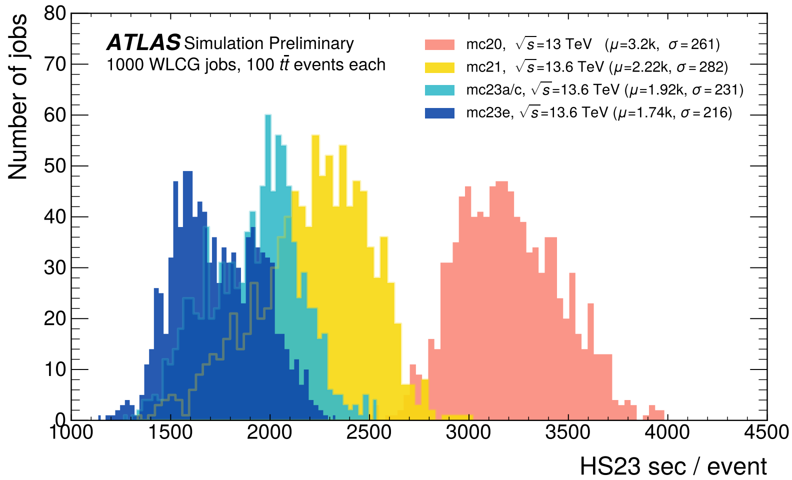

Figure 2: The mean CPU time per event simulated in Full Simulation comparing mc20 (Run2 MC Campaign) running at 13 TeV centre-of-mass energy (in red), mc21 (start of Run3 MC Campaign), running at 13.6 TeV centre-of-mass energy (in yellow), mc23c (Run3 MC campaign covering 2023 data taking), running at 13.6 TeV centre-of-mass energy (in light blue) and mc23e (Run3 MC campaign covering 2024 data taking), running at 13.6 TeV centre-of-mass energy (in dark blue) measured in standardised HS23 seconds [10] (mc20 uses GEANT4 10.1, mc21 and mc23x use GEANT4 10.6).

4. Possible future changes

Development of new optimisations continues for 2025 data-taking and beyond. These developments include the following:
Advanced Compiler Optimisations (Profile Guided optimisation) - This is the next step after LTO. It uses profile driven feedback to further optimise the big library.
High-η particle rejection - Extension of the Beam-pipe Killer. Kill primary particles generating secondaries close to the beam-pipe at $\sim 5-6$ m from the IP. Initial studies look promising.
Parameter Tuning of In-Field Tracking - Customise GEANT4 tracking parameters based on particle type, properties and position to optimise CPU performance without compromising precision.
Adoption of `G4HepEM` and specialised transport - A new compact GEANT4 EM library optimised

to be used for HEP EM showers development and transport. It provides significant speed-up compared to the general Geant4 EM library in stand-alone tests.
Adoption of `G4NeutronGeneralProcess` - A super-process for neutron physics.
Switching off Energy Loss fluctuations in Geant4[3] - This was found to be physics output-changing and so has been postponed until the next major MC campaign.
Re-implementation of EMEC Geometry - The as-built EMEC has a complicated "Spanish Fan" geometry. Efficient description using the `G4Solid` classes available in early versions of Geant4 was not possible. Therefore custom solids were used to implement the geometry algebraically. Simulation of particles in the EMEC is therefore particularly slow.[4] A new implementation based on data taken from the technical drawings and using `G4GenericTrap` solids has been developed. Initial tests in a stand-alone Geant4 example showed a factor of four improvement in tracking performance over the custom solid implementation. The next step is to assess the physics modelling with the new geometry implementation.

5. Summary

The ATLAS Simulation group aims to provide Monte Carlo samples with consistent physics for entire LHC runs for analysers. New optimisations which do not change the physics modelling are included between yearly sub-campaigns. Any optimisations which alter the physics modelling and any physics modelling improvements are included between campaigns. Multiple optimisations were introduced between the Run 2 (mc20) campaign and the latest Run 3 campaign (mc23e - matching 2024 data), increasing throughput by a factor of 1.84. There is a healthy programme of on-going development to include further optimisations in the future both from adopting new Geant4 features and by improving code within Athena. Further performance improvements are expected for the sub-campaign for 2025 data.

References

[1] ATLAS Collaboration: *Software and computing for Run 3 of the ATLAS experiment at the LHC*. https://arxiv.org/abs/2404.06335, 2024

[2] ATLAS Collaboration: The ATLAS Experiment at the CERN Large Hadron Collider. In: *Journal of Instrumentation* 3 (2008), Aug, Nr. 08, S08003. http://dx.doi.org/10.1088/1748-0221/3/08/S08003. – DOI 10.1088/1748–0221/3/08/S08003

[3] Agostinelli, S. et al.: GEANT4–a simulation toolkit. In: *Nucl. Instrum. Meth. A* 506 (2003), S. 250–303. http://dx.doi.org/10.1016/S0168-9002(03)01368-8. – DOI 10.1016/S0168–9002(03)01368–8

[4] ATLAS Collaboration: *Athena*. https://doi.org/10.5281/zenodo.2641997

[5] ATLAS Collaboration: *ATLAS Geant4 Performance Optimization Plots*. https://atlas.web.cern.ch/Atlas/GROUPS/PHYSICS/PLOTS/SIM-2019-001/,

[3] Geant4 command `'/process/eLoss/fluct false'`

[4] The use of custom solids also blocks the implementation of a GPU-based simulation of ATLAS in Geant4.

[6] Wenzel, Sandro ; Apostolakis, John ; Cosmo, Gabriele: A VecGeom navigator plugin for Geant4. In: *EPJ Web Conf.* 245 (2020), 02024. http://dx.doi.org/10.1051/epjconf/202024502024. – DOI 10.1051/epjconf/202024502024

[7] Woodcock, E.: Techniques used in the GEM code for Monte Carlo neutronics calculations in reactors and other systems of complex geometry. In: *Proceedings of the conference on applications of computing methods to reactor problems, 1965* 557 (1965). https://cir.nii.ac.jp/crid/1573668925582592640

[8] Hahnfeld, Jonas ; Ivanchenko, Vladimir ; Novak, Mihaly ; Pandola, Luciano ; Sawkey, Daren: Geant4 electromagnetic physics for Run3 and Phase2 LHC. In: *EPJ Web of Conf.* 295 (2024), 03018. http://dx.doi.org/10.1051/epjconf/202429503018. – DOI 10.1051/epjconf/202429503018

[9] Novak, Mihaly: *G4HepEM*. https://g4hepem.readthedocs.io/en/latest/index.html,

[10] ATLAS Collaboration: *Run 3 Full Simulation CPU time mc23e*. https://atlas.web.cern.ch/Atlas/GROUPS/PHYSICS/PLOTS/SIMU-2023-06/,