ATLAS Full-simulation optimisation for HL-LHC

Workshop sul Calcolo nell'INFN: Paestum, 23 - 27 maggio 2022

Caterina Marcon on behalf of the ATLAS Collaboration

INFN MILANO
Istituto Nazionale di Fisica Nucleare
Sezione di Milano

ATLAS EXPERIMENT

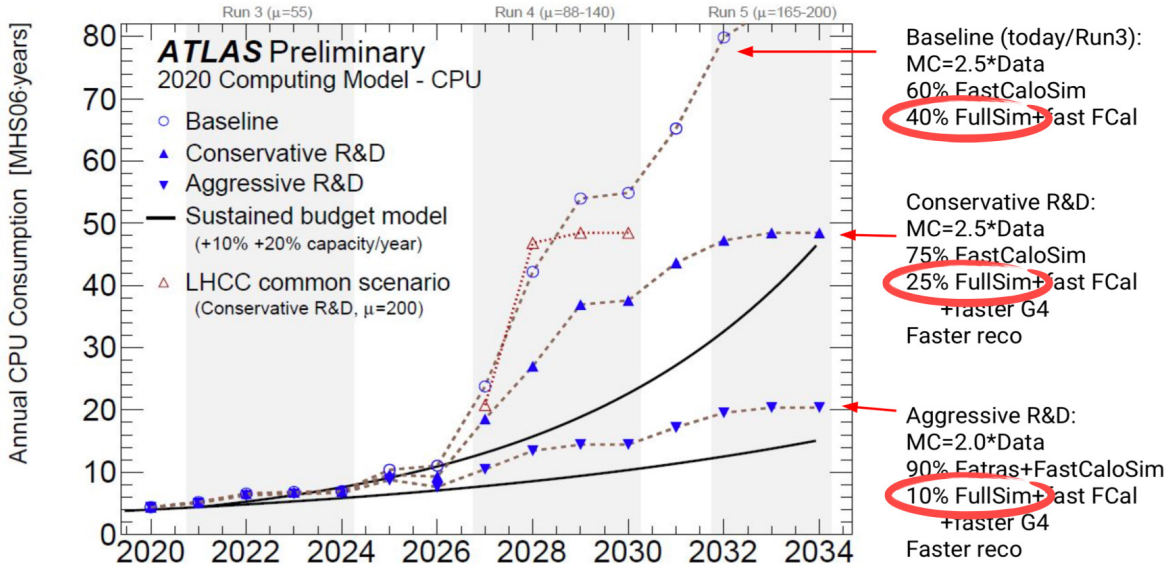
Outline

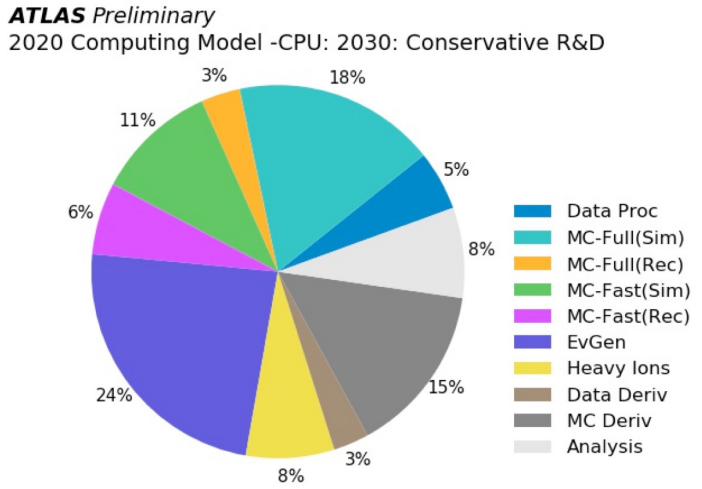
- Computing complexity challenge;
- Geant4 Optimization Task Force;
- Intrinsic Geant4 improvements;
- Geometry implementation improvements;
- Reducing Operations;
- Compile-time optimizations;
- Physics optimizations;
- Conclusions.

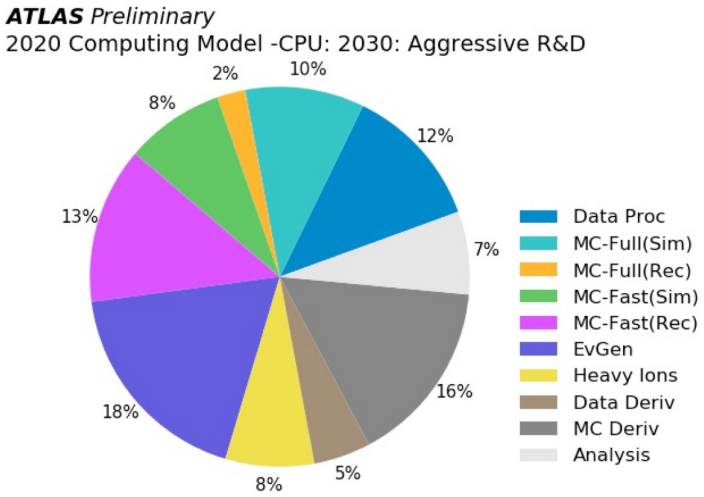
Computing complexity challenge

- The **upgrade for Run 3 and beyond** represents a step change for ATLAS: the event rates will be approximately **10 times larger** than during previous runs;

- Accurate simulations and larger Monte Carlo samples will be needed to achieve the desired precision in physics measurements while avoiding that simulation dominates the systematic uncertainties.







Geant4 Optimization Task Force

- The **Geant4 Optimization Task Force (G4TF)** is responsible for optimizing the performance of the ATLAS G4 simulation software:
  - Investigating configuration options and simplified geometries and magnetic-field descriptions;
  - Improving the ATLAS interface code to G4.
- The TF mandate is to achieve for Run 3 > **30% CPU performance improvements** with respect to the Run-2 simulation.

Intrinsic Geant4 improvements

Newer Geant4 versions & calorimeter Test Beam Integration

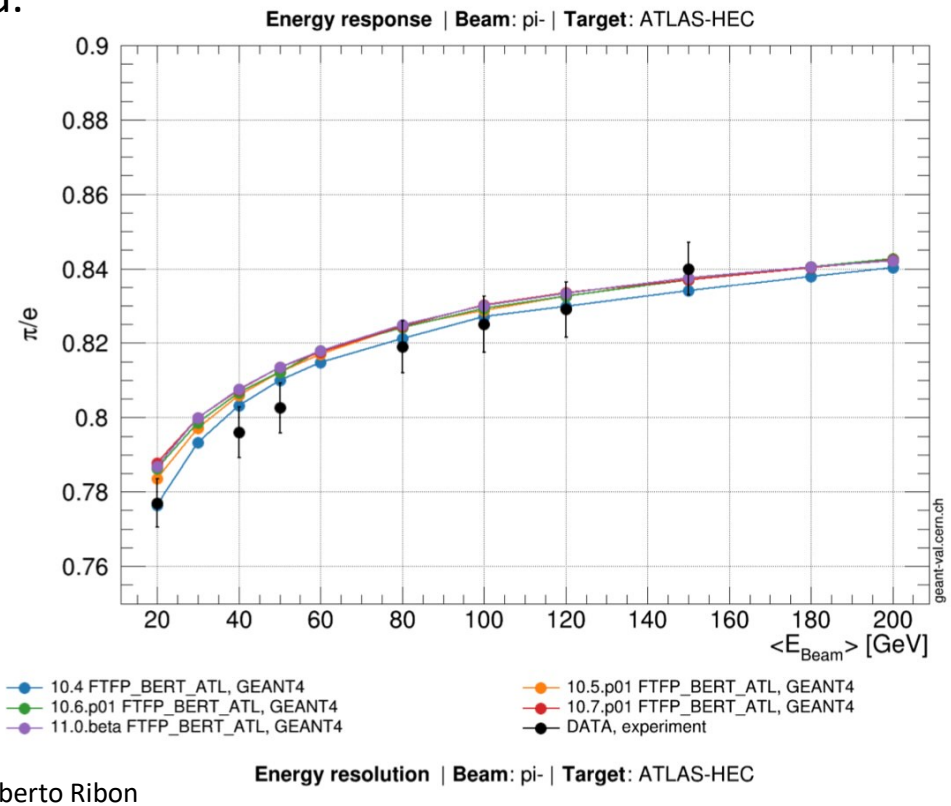
- For Run 2 samples Geant4 10.1.patch03.atlas07 has been used in x86_64-centos7-gcc62-opt platform;

- For Run 3 ATLAS considers two versions in x86_64-centos7gcc11-opt (C++17) platform:

  - **Geant4 10.6.patch03.atlas03** (first samples for calibration): all the optimizations presented have been implemented using this version;

  - **Geant4 10.7.patch02.atlas01**: in the validation phase some unexpected discrepancies were found (e.g. jet EM fraction and increase in constituents, jet response decrease) which are under investigation.

- New Geant4 versions must be validated both with respect to the previous versions and with respect to the data;
- A collaboration between Geant4 and ATLAS allows to automatically validate Geant4 using hadronic and electromagnetic calorimeters test-beam data [1];
- So far, Hadronic Endcap Calorimeter (HEC) ATLAS data has been considered.



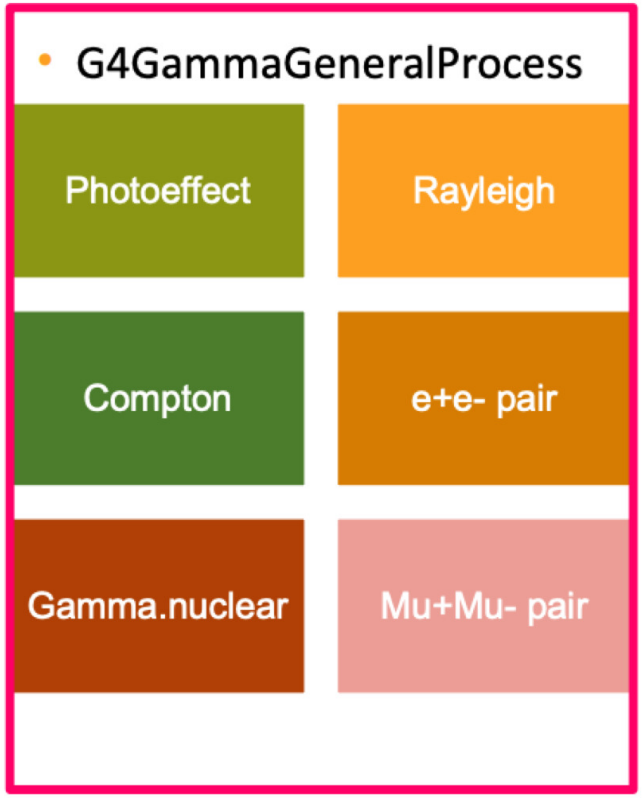
[1] Lorenzo Pezzotti & Alberto Ribon

Gamma General Process

- **G4GammaGeneralProcess** has been back-ported from G4 10.7;

- It is a **super-process** incorporating all the physical processes involving photons, thus allowing the SteppingManager to see only one physics process -> reduced number of instructions;

- Tests carried out considering 100 ttbar as primary events underlined a speed up of **4.3%**.



VecGeom

- VecGeom [1], the **vectorized geometry library** for particle-detector simulation, is a geometry modeller library with hit-detection features;
- VecGeom has some promising features:
  - Build a hierarchic detector geometry out of simple primitives and use it on the CPU or GPU(CUDA);
  - Collision detection and navigation in complex scenes;
  - SIMD support in various flavours:
    - True vector interfaces to primitives with SIMD acceleration when beneficial;
    - SIMD acceleration of navigation through the use of special voxelization or bounding box hierarchies.
  - VecGeom also compiles under CUDA.

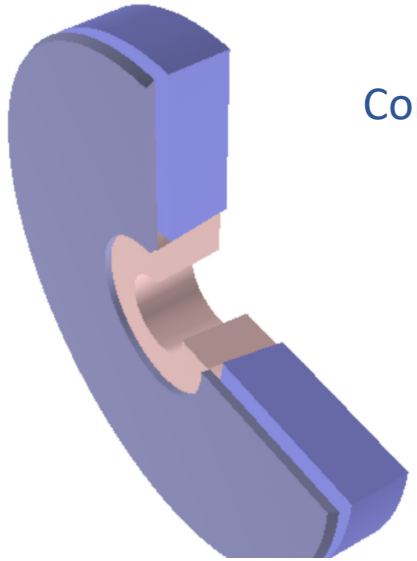
- The speed up given by the implementation of VecGeom in ATLAS is promising (overall speed up **~ 6%**) but, at the moment, it gives some problems in the execution phase of the simulation: ~ 10% of the events are aborted -> under investigation.

[1] https://gitlab.cern.ch/VecGeom/VecGeom

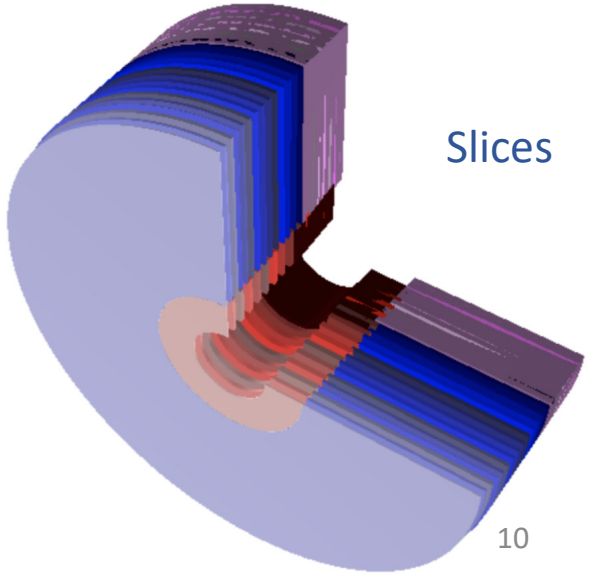
Geometry implementation improvements

Simplifying EMEC Geometries reducing G4Polycone usage

- EMEC geometry is currently described by a **custom Geant4 solid** using **G4Polycone**;
- G4Polycone is **slow** because it defines a complex shape based on polycones -> **complex routines** that describe the geometry;
- Two alternative and simpler shapes have been tested:
  - **Cone:** improved shape using G4ShiftedCone: outer wheel divided into two conical-shaped sections;
  - **Slices:** new LArWheelSliceSolid: each wheel is divided into many thick slices along the Z axis.
- Slices provided **5-6% speed up**.



Cone



Slices

GPU-Friendly EMEC Implementation

- The ultimate goal is to have an EMEC implementation based on **standard Geant4 shapes and GPU-friendly** [1];
- So far, the accordion shape is not available within the GEANT4 standard geometry shapes;
- Two alternative shapes, able to replicate an extremely simplified EMEC geometry, have been investigated:

Geometry	Time (s)
G4Trap	111.67
G4GenericTrap [2]	72.07

[1] https://indico.cern.ch/event/1052654/contributions/4525306/attachments/2310908/3932523/AdePT%2026th%20Geant4%20Collaboration%20Meeting.pdf

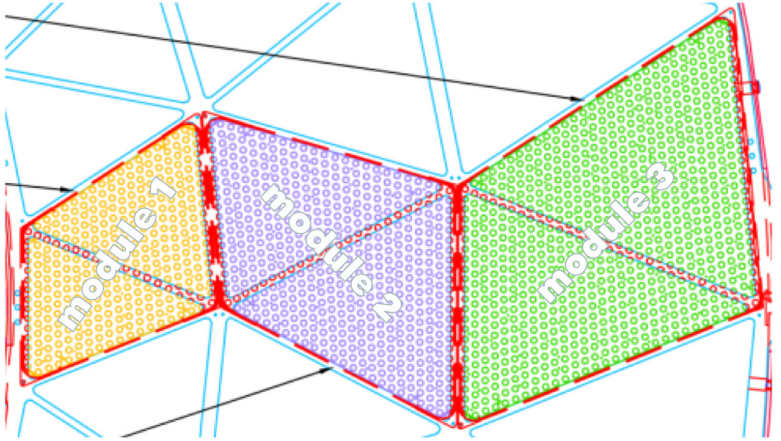
[2] Converted from G4TwistedTrap.

TRT Geometry Optimization

- The current implementation takes advantage of **boolean solids**: two triangular prisms are merged together by their common face;

- This approach is not optimal as Boolean operations are **slow** and they can cause **tracking issues** especially in presence of coincident surfaces;

- Describe these volumes using alternative shapes:
  - arbitrary trapezoid (**Arb8**);
  - the Boundary REPresentation (**BRep**).



96 trapezoidal modules grouped in 3 types characterized by an increasingly larger cross sectional area

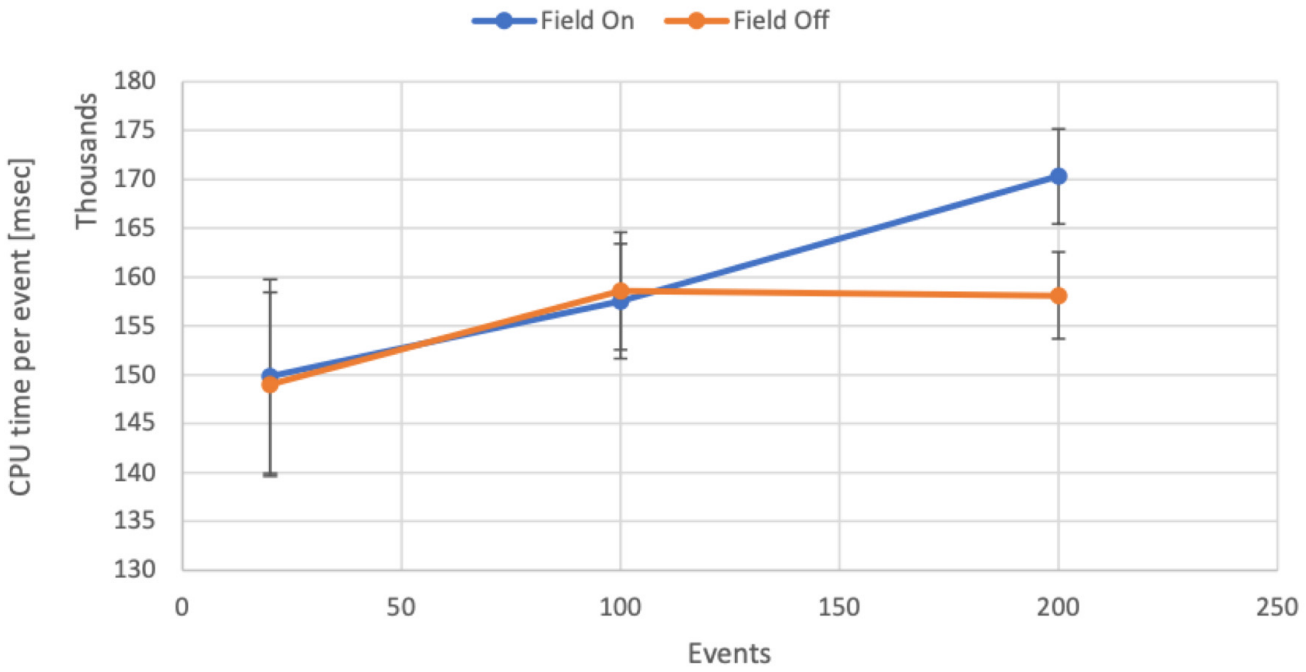
- A speed up of 1.5% is observed for the Arb8 representation:

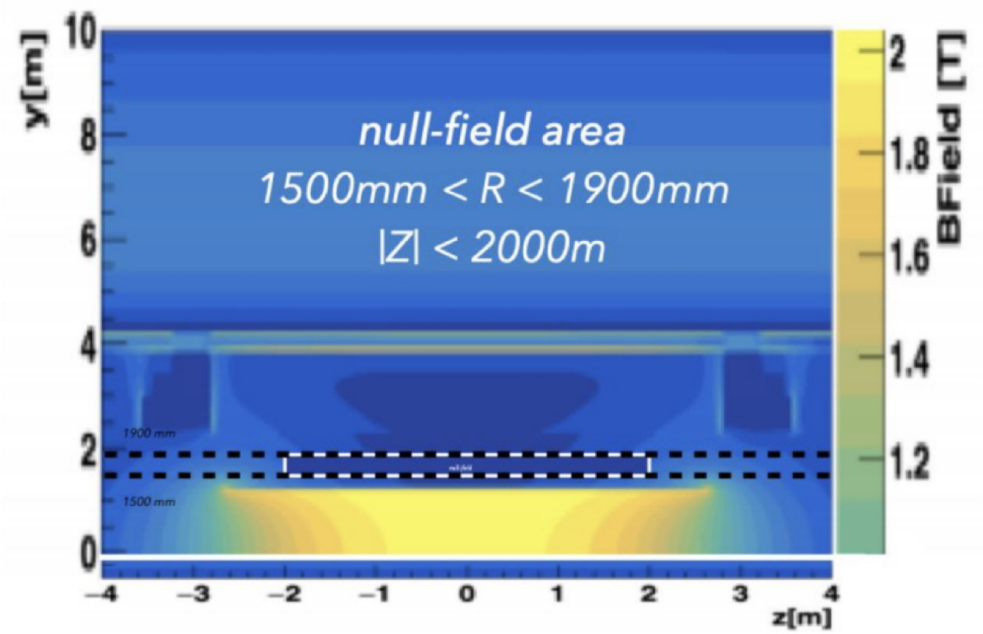
Module shapes	Execution time (s)	Improvement
Boolean solids	1663	Reference
Arb8	1638	+1.5%
BRep	1675	-0.7%

Reducing Operations

Magnetic Field Tailored Switch-off

- Speed up observed when **switching-off magnetic field** in LAr calorimeter (except for muons) without affecting shower shapes:
  - ~3% speed up for full ttbar events;
  - ~7% speed up for 1GeV e- on 0<η<0.17.
- Possibility to extend solution to other detector regions.

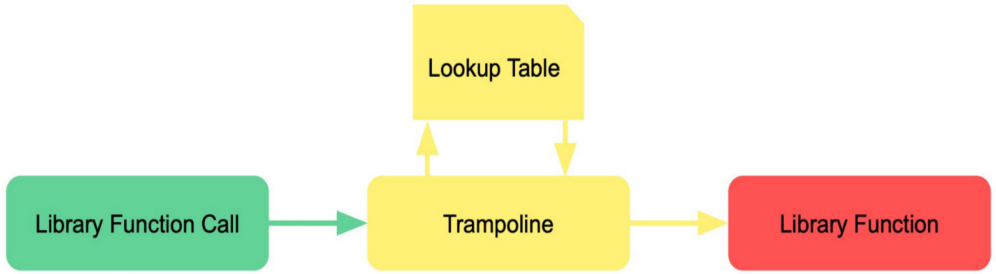


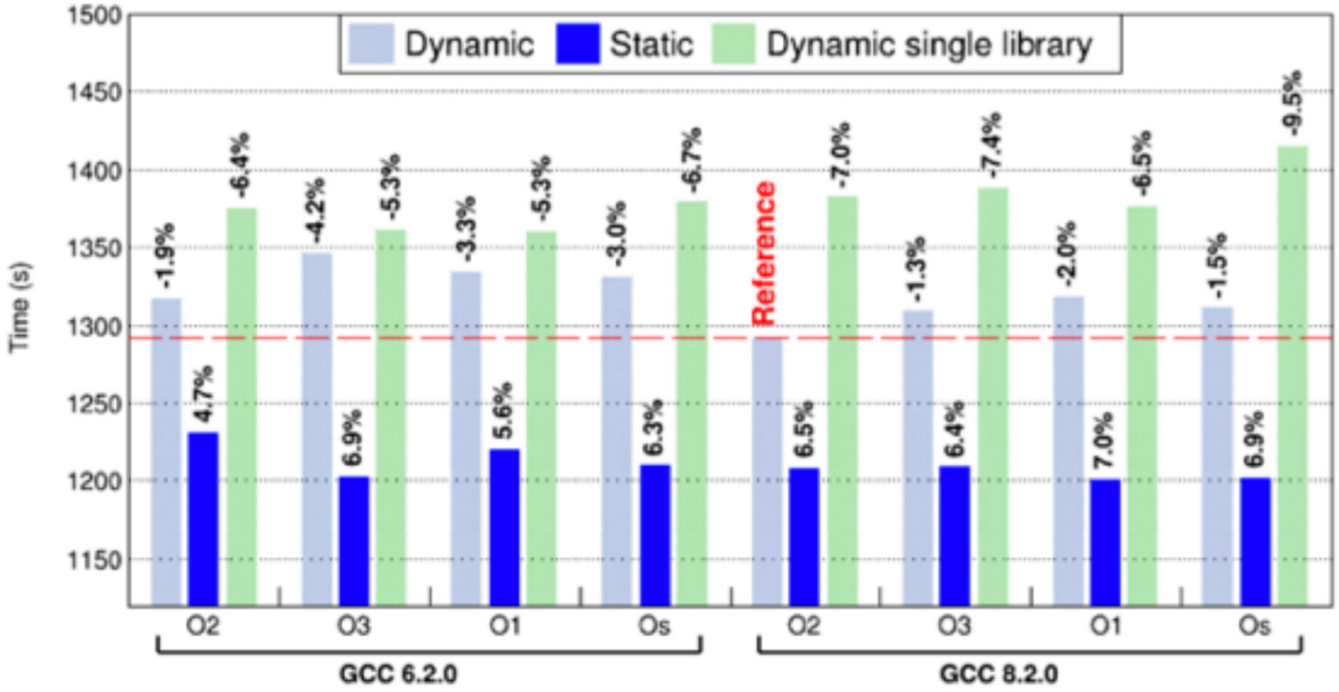


Compile-time optimizations

Big static library

- Tests with a **big dynamic library** (groups of all libraries from packages that use Geant4) showed a 10% slowdown due to trampoline effect;





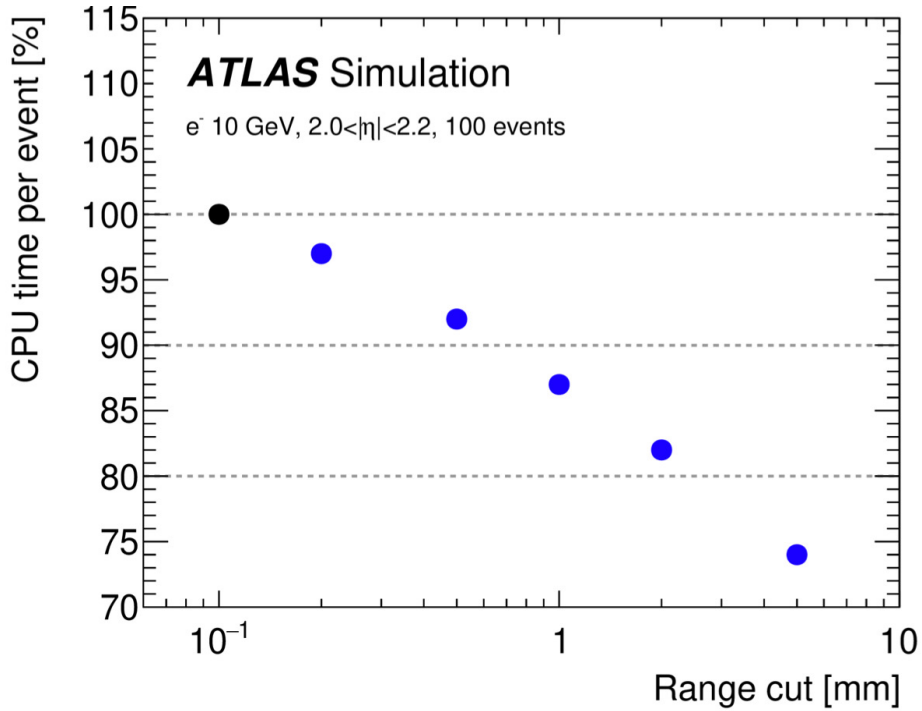
- **Big Geant4 static library** has been implemented and a average speed up of 6-7% has been showed;

- No validation needed.

Physics optimizations

EM Range Cuts

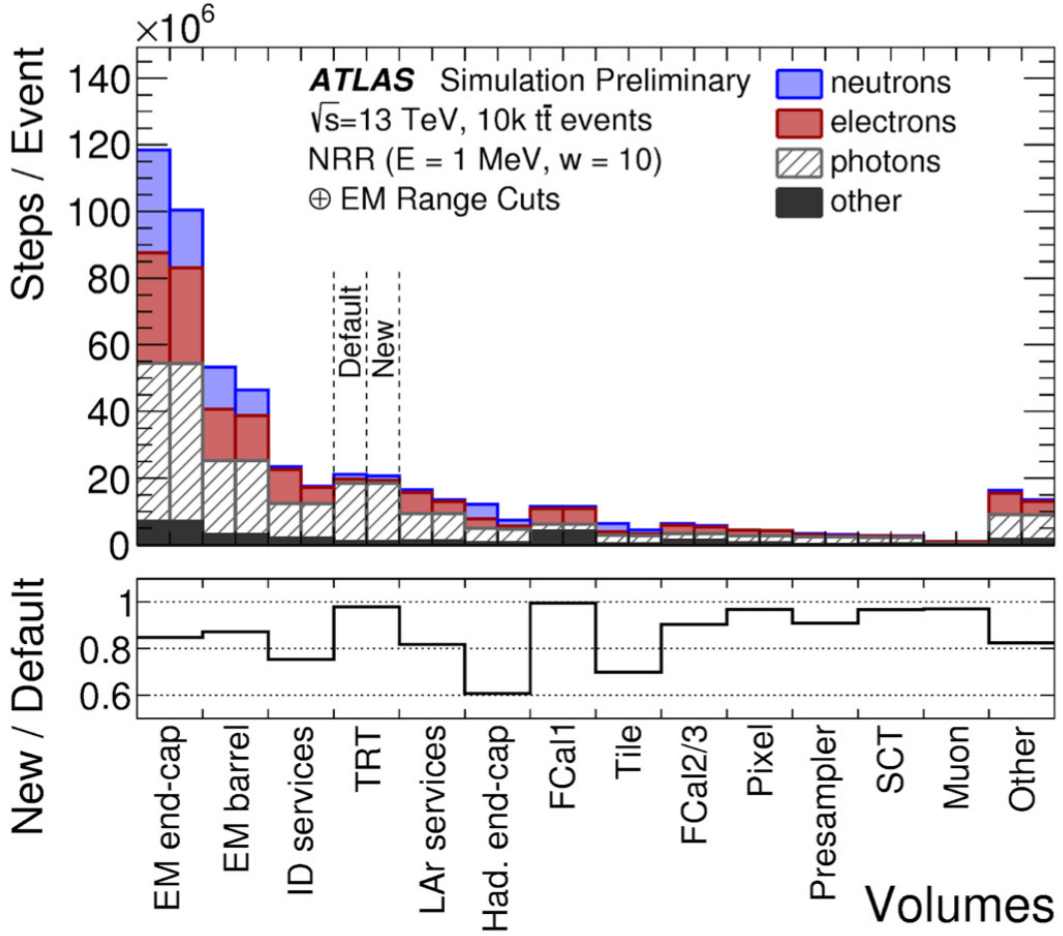
- **Increased range cuts** can reduce the number of photons, thus reduce the transportation steps and increase computational performance
- OFF by default for three processes: Compton, conversion, photo-electric effect;
- Turning them on provide ~**6-7% speed up**;



- Side-effect: high range cuts can degrade the accuracy of the simulation (e.g. shower shape);
- **Machine-Learning-based correction** can be applied as a post-processing step using batch processing and accelerator hardware;
- ML inference time negligible compared to simulation time reduction.

Russian Roulette

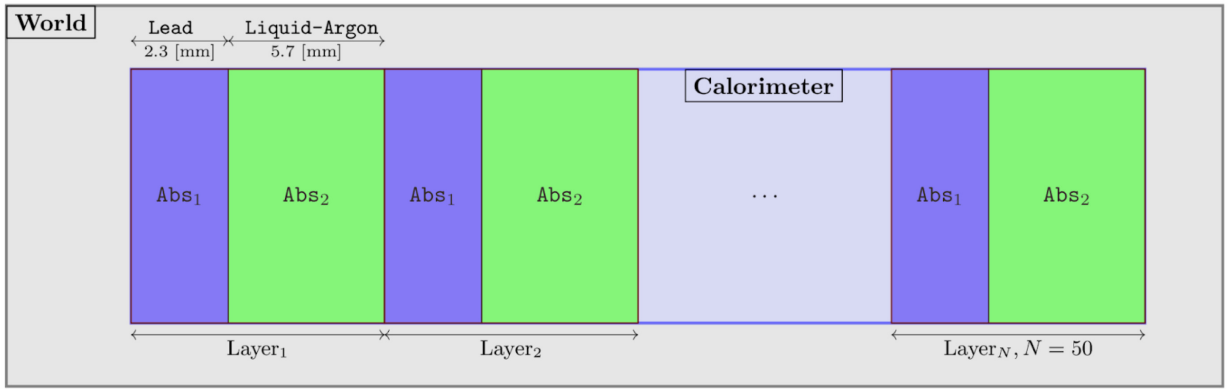
- **Neutrons and photons** take majority of CPU time;
- Photon/Neutron Russian Roulette (PRR/NRR): randomly discard particles below energy threshold and weight the energy deposits of remaining particles accordingly;
- NRR performance: **10%** speed up with 2 MeV threshold for neutrons.



Woodcock Tracking

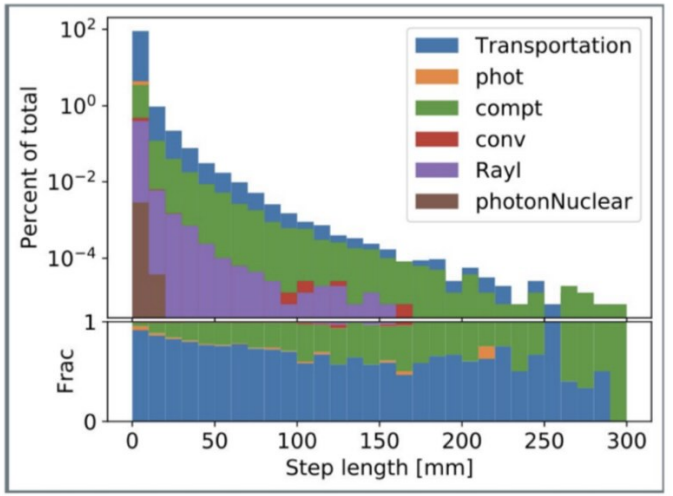
https://www.sciencedirect.com/science/article/pii/S0306454916303498

- The foundation of this approach:
  - Performs tracking in geometry considering **only one material**: the densest (e.g. Pb);
  - Interaction probability is proportional to the cross section ratio between the real material and the densest.
- Avoids many steps due to the boundaries (Transportation) since there are no boundaries -> especially powerful in highly granular detectors;
- **Preliminary study** using simplified layered Pb/LAr calorimeter showed up to 10% computational speed.

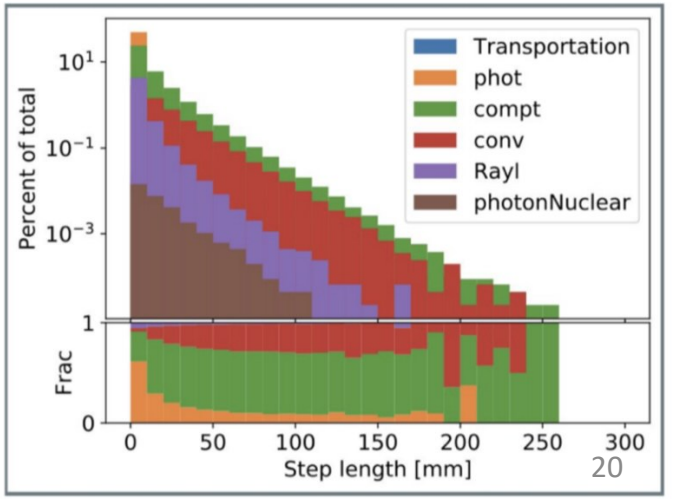


- Implementation for tha ATLAS EMEC is ongoing;

EMEC Total



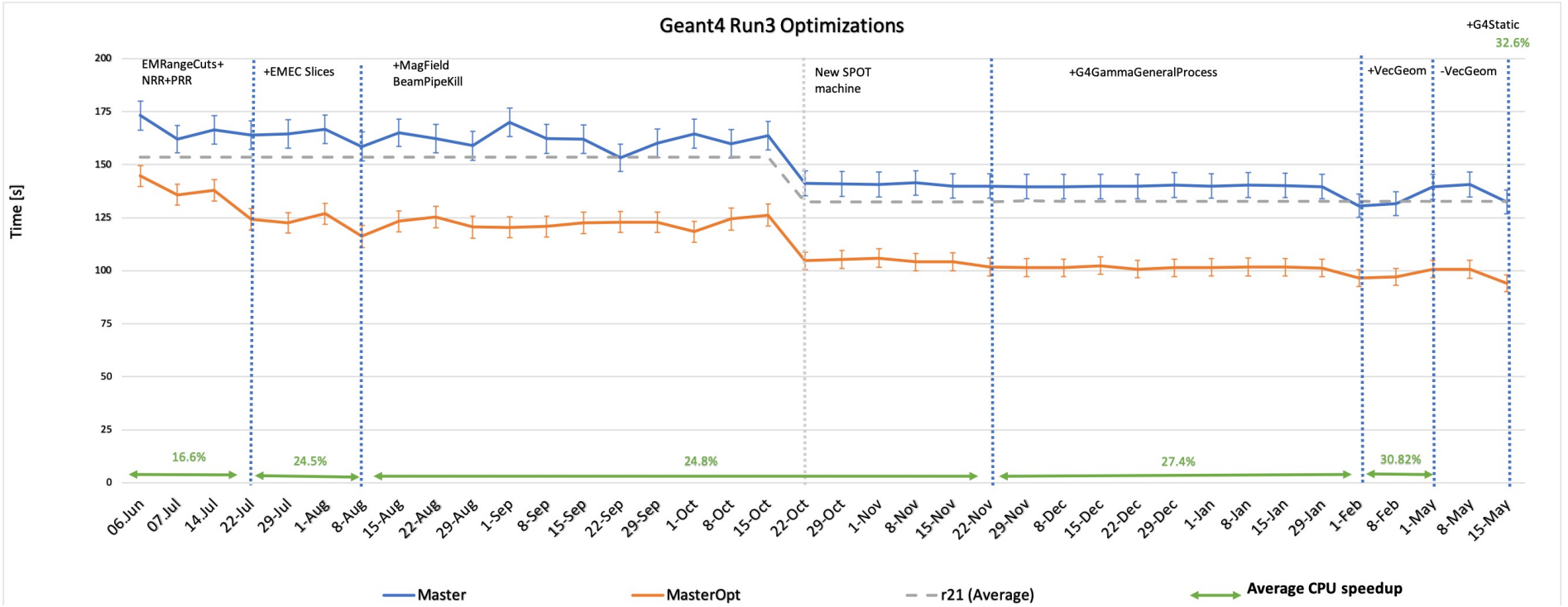
EMEC Lead Only



Conclusions

Conclusions

- During the High-Luminosity LHC phase, event rates will be approximately 10 times larger than during previous runs;
- In ATLAS, an **active R&D program** is ongoing to reduce the time spent for simulations by optimizing the Geant4 CPU and memory footprints;
- So far, more than **32% CPU speed up** has been reached: it corresponds to **+48% throughput** using the same computational resources.



Thank you for the attention

Backup

Calorimeter Test Beam Integration

- Workflow:

1. Port the ATLAS HEC simulation into a new standalone Geant4 simulation;

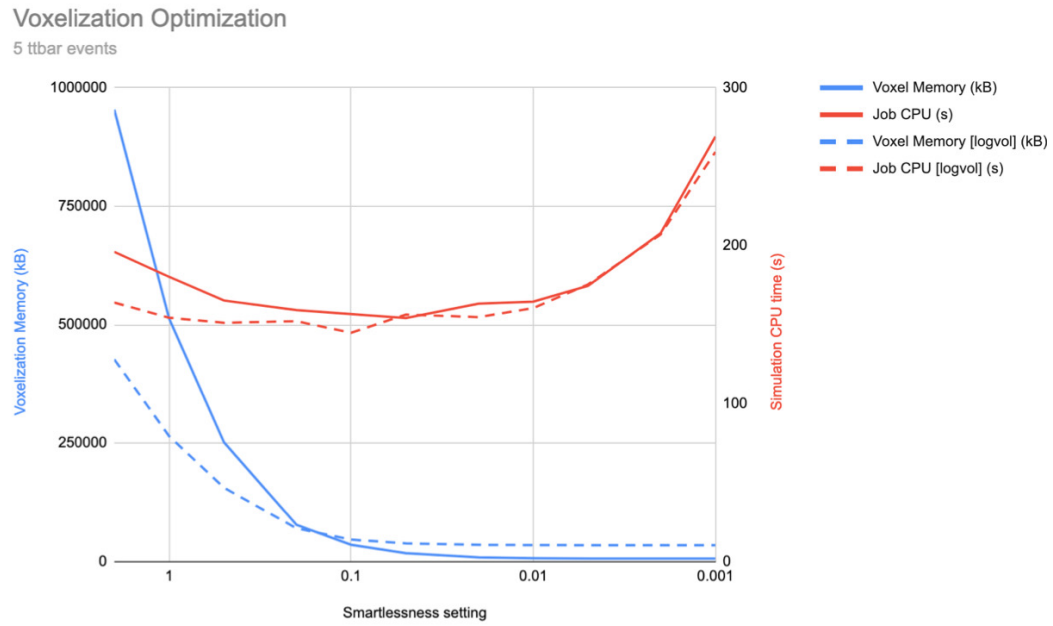
2. Perform Geant4 validation against the ATLAS HEC testbeam data;

3. Porting the application into the Geant Val testing suite.

Outlook

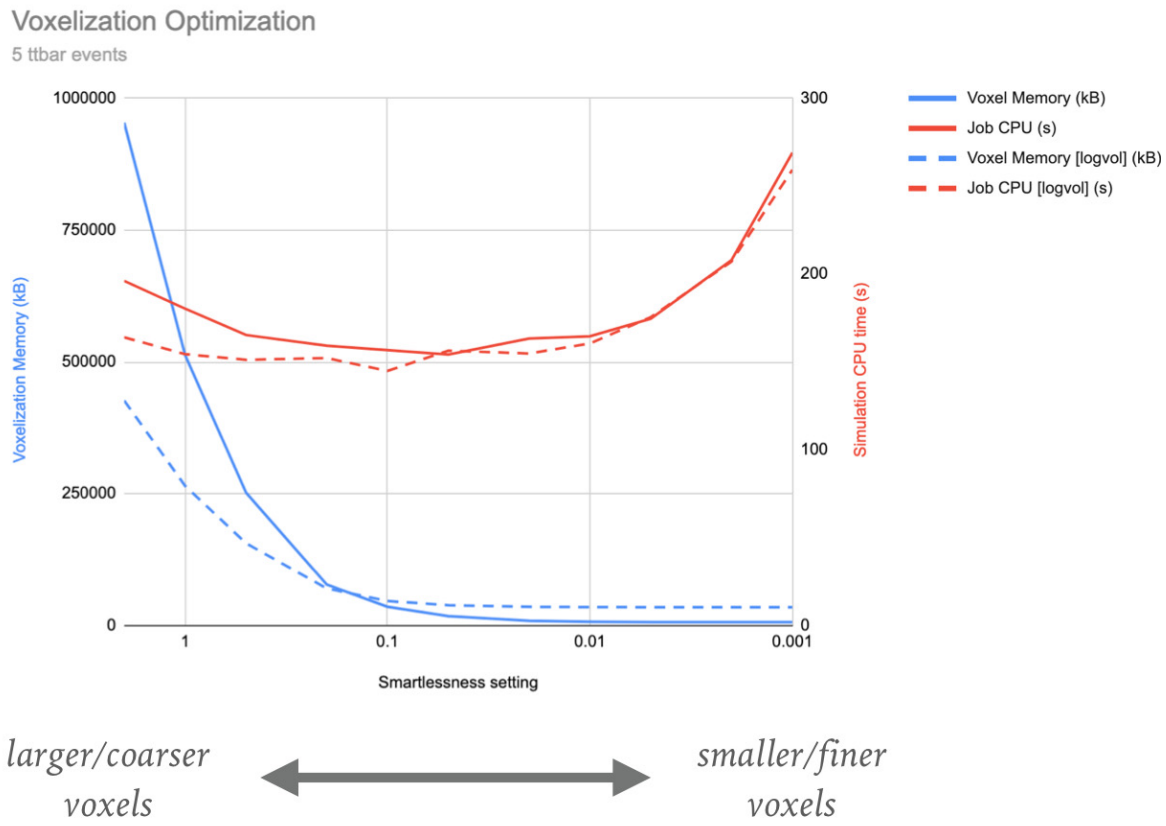
- Beyond the optimizations described, more optimizations/improvements are upcoming:

  - Effort to reduce the Thread Local Storage usage in Athena and Geant4;
  - Voxel Density Tuning: Optimize the values of Smartless parameter for a balance between memory used for the detector description and CPU time for simulation.



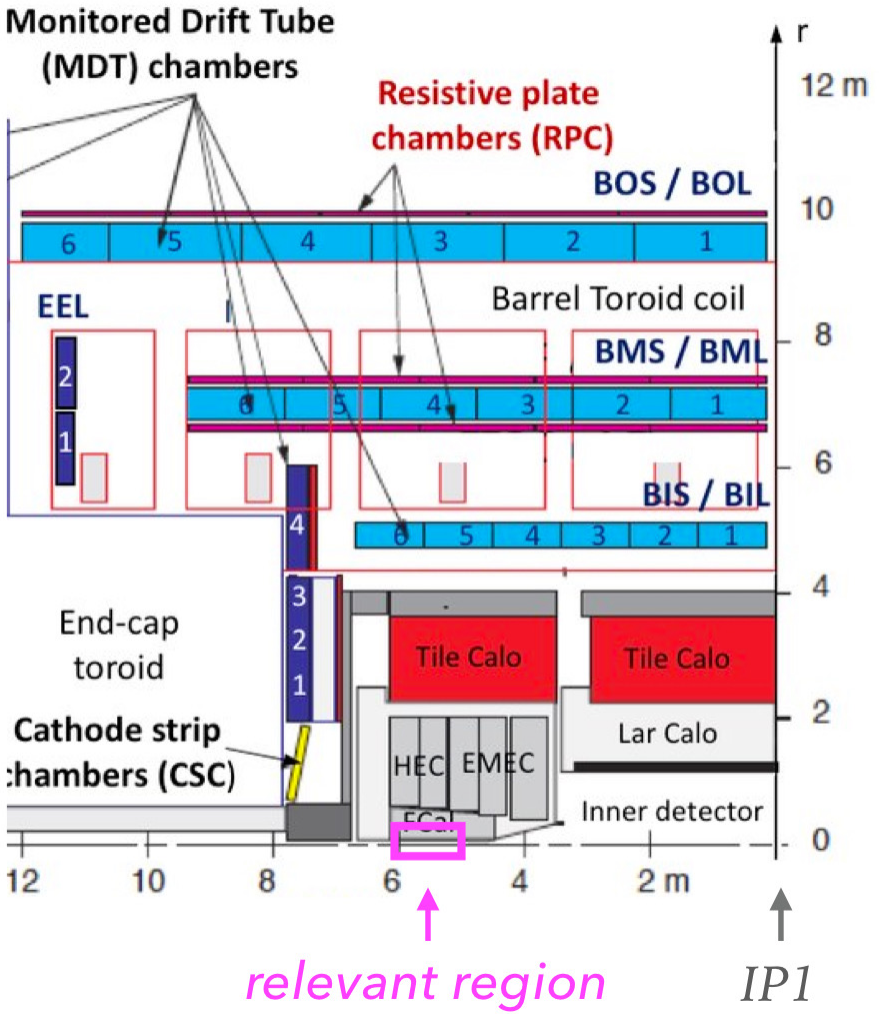
Voxel Density Tuning (to be implemented)

- Tracking can be optimized by voxelization, the size/granularity of the voxels can be tuned by the Smartless parameter;
- Goal: Optimize the values of Smartless parameter for a balance between memory used for the detector description and CPU time for simulation;
- Simulation accuracy should also be checked (although no effect is expected).



Outlook

- New particle filter implementation:
  there is a huge amount of secondaries being created 5-6m away from (0,0,0) that will never cause any energy deposit in the calorimeters or a muon hit;
- The primary particles generating these secondaries could just be dropped directly:
  - all particles at η>6 are already killed;
  - apply the same approach to:
    - particles at η>5 and pT < 10 GeV;
    - or/and particles at η>4 and pT < 1 GeV.



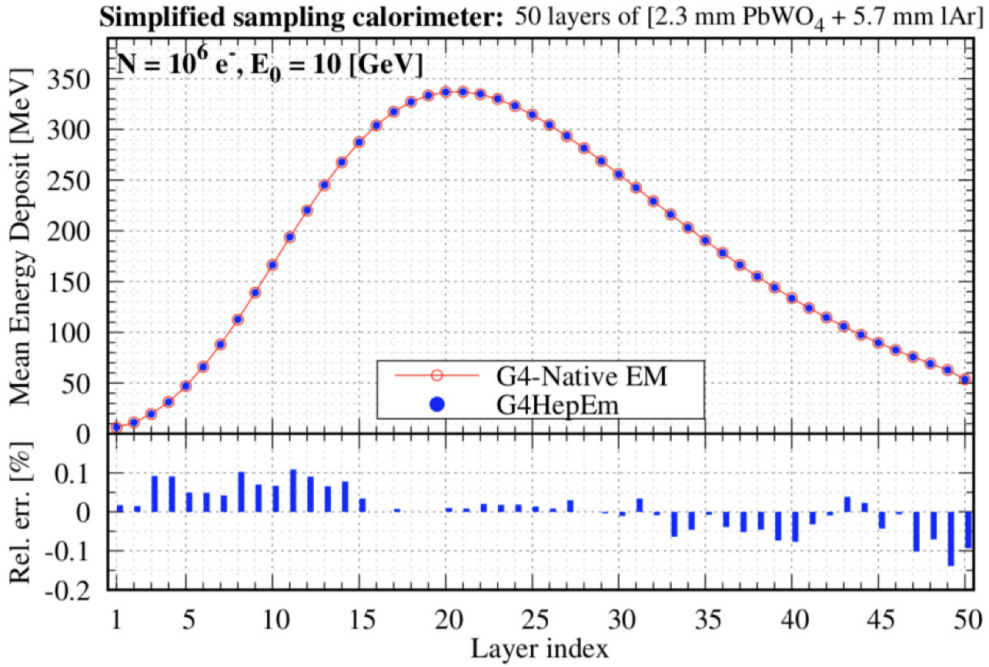
New particle filter (to be implemented)

- Workflow:

1. generate a large sample of single particles with $4{,}5 < |\eta| < 6$ and different energies;
2. map out which η and E combinations can produce a relevant signal;
3. drop the rest directly with a new particle filter;
4. Approach similar to Russian Roulette (see slide 20).

G4HepEM Library Integration

- G4HepEM library is a new compact Geant4 EM library [1];
- Optimized to be used for HEP electromagnetic showers development and transport;
- More compact and GPU-friendly;
- It has been already integrated in FullSimLight and Athena but some performance plots have showed some discrepancies -> under investigation;



CMS detector configuration simulating ttbar events

	Physics List	Specialised Tracking	difference
G4NativeEm	2889 s	2747 s	-4.9 %
G4HepEm	2847 s	2660 s	-6.6 %
difference	-1.5 %	-3.2 %	-7.9 %

Note: significant performance gain due to the specialised tracking of e^-/e^+ and γ even already using GEANT4 native processes that is boosted further with G4HepEm (even in its current, preliminary phase)

max 0.1% change in simplified calorimeter observables

[1] https://indico.cern.ch/event/1052654/contributions/4524767/attachments/2309218/3929219/G4HepEm_SpecTracking_MNovak.pdf

Impact of static libraries on working nodes

- Static libraries result in larger executables;
- this means more data for the loader to copy into the memory of the working machines;
- this could result in a penalty due to the limited speed of data connections (e.g. network)

However:

- the overall size of the executable plus all dynamic libraries is the same as the big static executable;
- Eventually, the amount of data to load is the same;
- What is instead impossible with a static build is to have pre-distributed copies of the libraries to the working nodes.

The anomaly of the big dynamic library

- Compared to the standard dynamic build, the 10% degradation of performance was not expected;
- The matter is currently being discussed by several experts of the simulation and IT departments, but a conclusive explanation has not yet been found;
- A slow-down due to the "trampoline/lookup tables" mechanism used to make calls between Geant4 and the simulation code is considered an option.
- However, the ongoing tests no longer involve this option and instead are focusing on the big static library.

EMEC accordion shape

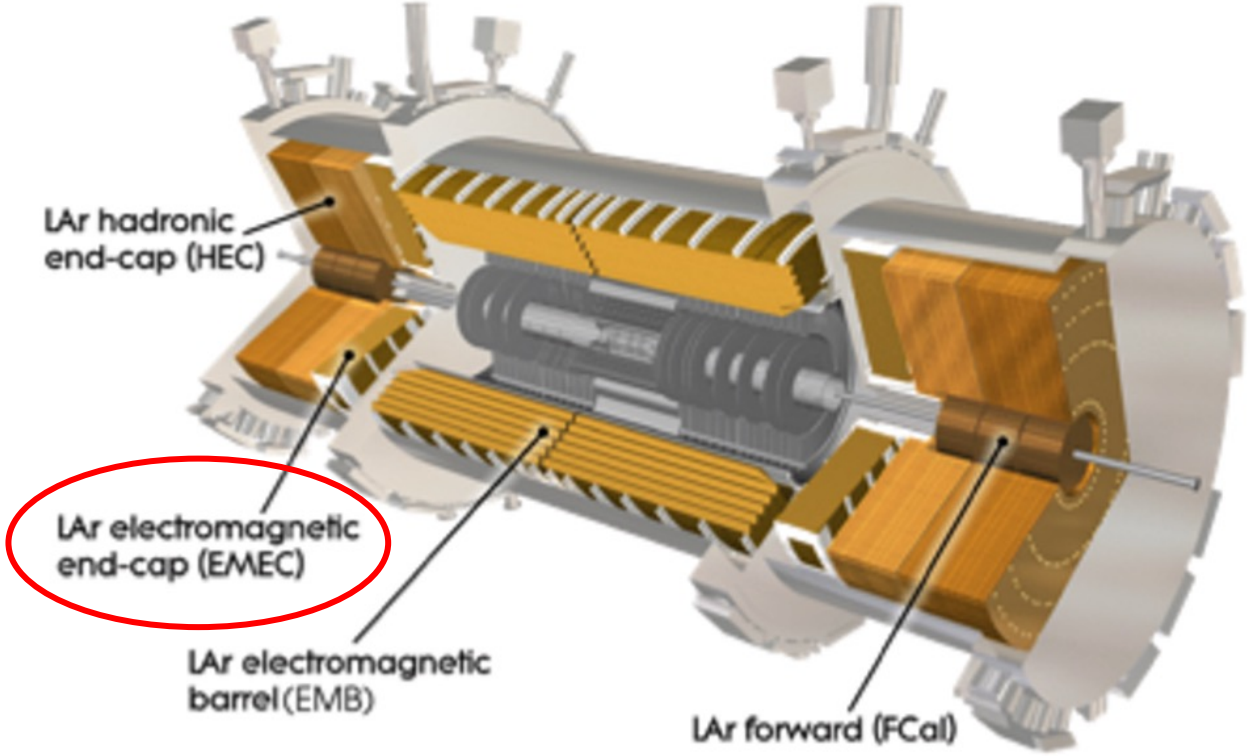


Fig. 1. The Atlas liquid Argon Calorimeter system.

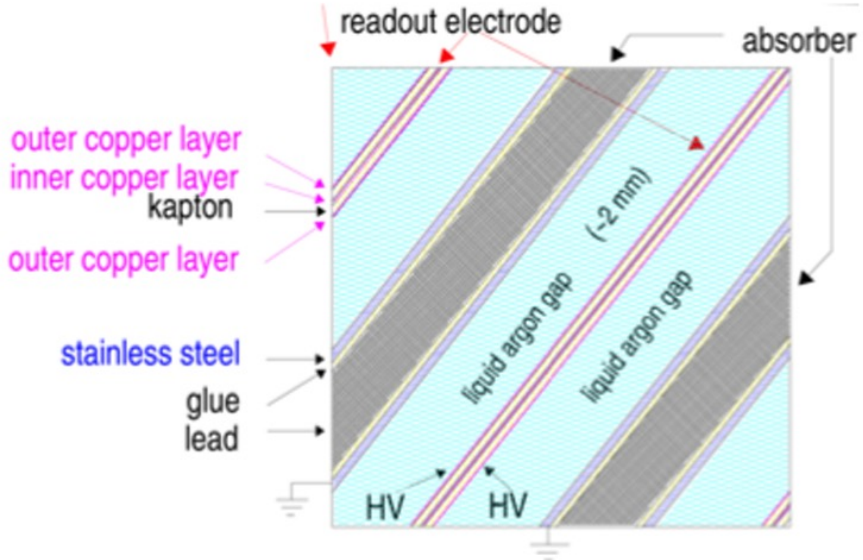


Fig. 2. Structure of the electrodes of the LAr calorimeter (accordion shape)