ATLAS Full Simulation Update

Walter Hopkins on behalf of the Full Simulation Group

Geant4 Technical Forum

April 6th, 2023

INTRODUCTION

Goal of ATLAS Full Simulation group: improving the computational performance of FullSim without significant loss of physics accuracy

Working closely with Geant4 experts on various topics

TOPICS:

- General bug fixes
- Geometry optimization
- Voxel density optimization
- ISF Particle killer
- Advanced Compiler Optimizations
- Geant4 11 in Athena
- Feedback to Geant4

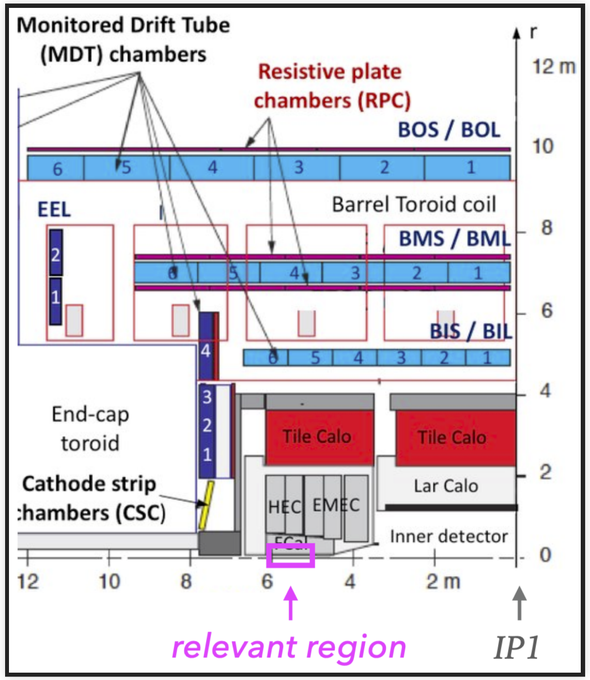
Only highlighting updates since previous report

GENERAL BUGS

- Bug in ATLAS version of Monopole Geant4 physics extension
  - Failed jobs when simulation of highly ionizing particles (q-balls, monopoles, etc)
  - Related to assumption of one global G4FieldManagers but ATLAS switches field managers
  - Now fixed due to work from John Apostolakis, John Chapman, and Mustafa Schmidt
  - Some of the code improvements from the fix will be ported to the Monopole G4Example by John Apostolakis
- Negative energy neutrons: generated when the cascade interface is called, causing a had0006 exception
  - Mihaly Novak and Alberto Ribon investigated and fixed this
  - Temporary fix: catch exception when there is a secondary with negative energy and abandon the interaction
  - Proper fix being worked on
- Index out of range error
  - Seems related to G4MultiBodyMomentumDist::passVerbose(int)
  - Mihaly and Alberto are also working on a fix for this

ISF PARTICLE KILLER

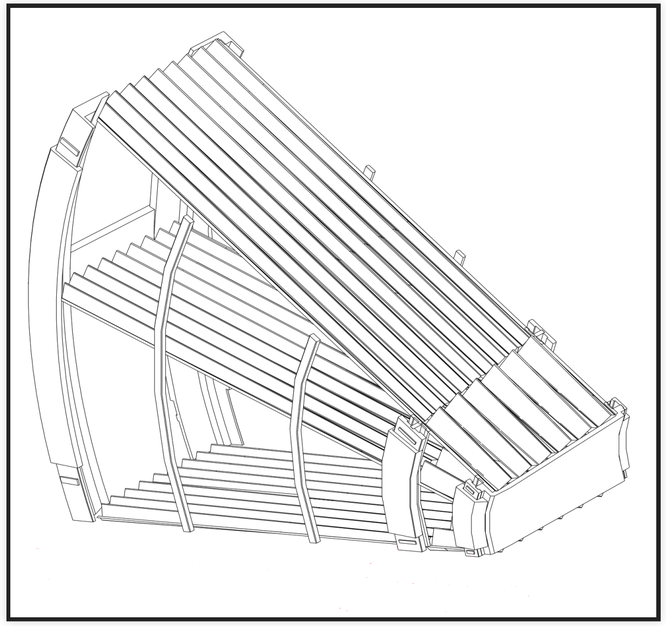
Goal: kill primary particles that mainly produce secondaries close to the beam pipe at 5-6 m



- Large number of particles are produced 5-6m from origin and close to the beam pipe
  - Equivalent to $|\eta| > 4.5$ but we already discard particles with $|\eta| > 6$
- Many of these particles won't deposit energy in calorimeters
- **Particles that originate these showers could simply be dropped to improve simulation efficiency**
- Map in η and E have been created and are being studied to find cuts to remove particles
- Work done by Dongwon Kim and Michael Dührssen

EMEC GEOMETRY OPTIMIZATION

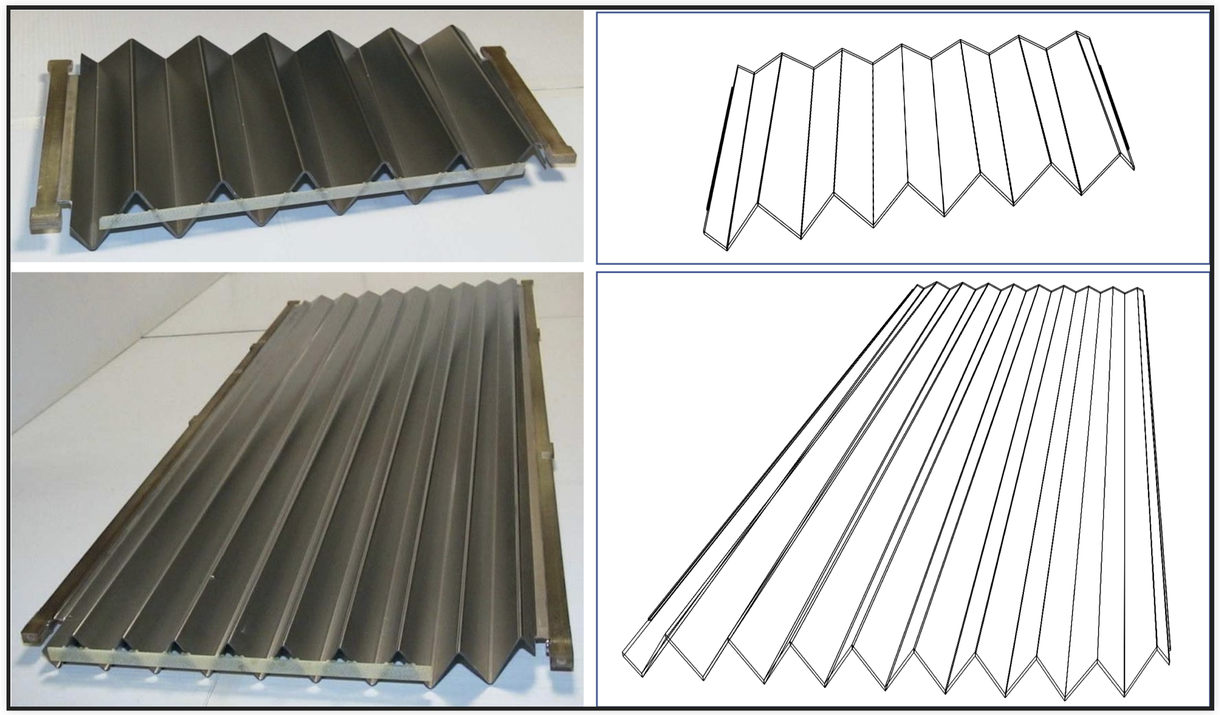
Goal: optimize Electro Magnetic End Cap (EMEC) calorimeter geometry to improve simulation speed



- EMEC complex geometry (accordion) is dominant computational resource user
- EMEC is currently described by custom solid which is not portable to GPUs
- Effort started by Akanksha Vishwakarma and continued by Evgueni Tcherniaev

EMEC GEOMETRY WITH G4SOLIDS

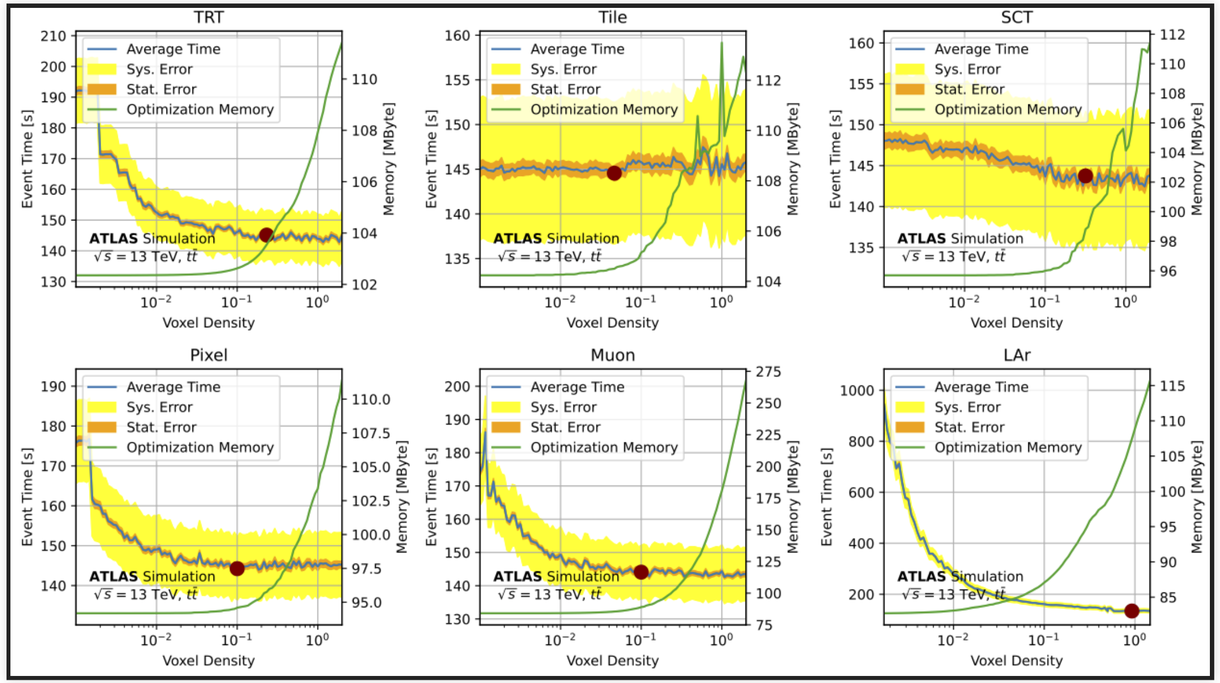
- Materials taken from custom solid C++ code
- Geometry based on technical drawings
- Some small discrepancies between TDR and current EMEC implementation have been noticed.
- Next steps: test CPU performance of different implementations
  - Without any subdivisions, with z-slicing, z-slicing and φ sectors, and potentially with the addition of division in r



VOXEL DENSITY OPTIMIZATION

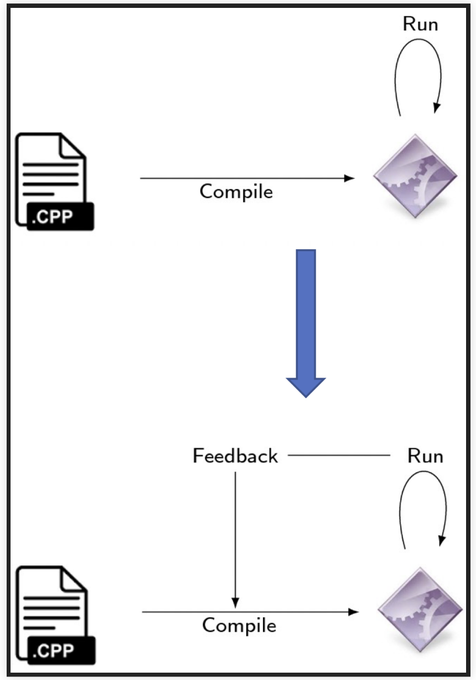
Goal: reduce memory consumption or improve CPU by changing voxel size/granularity

- Tune the voxel size and granularity via "smartless" G4 paramter for optimal memory and CPU consumption
- Initially found that voxel density can be tuned to decrease of memory usage without CPU consumption change
  - Later studies found a significant increase in CPU consumption with this setup
  - Suspected cause: change in geometry setup (no VecGeom vs VecGeom)
- Overall memory consumption actually not high → can we gain some CPU performance at the cost of memory usage?
  - Will investigate for LAr (dominant by a large margin, see left y-axis)
- Work done by Mustafa Schmidt



ADVANCED COMPILER OPTIONS

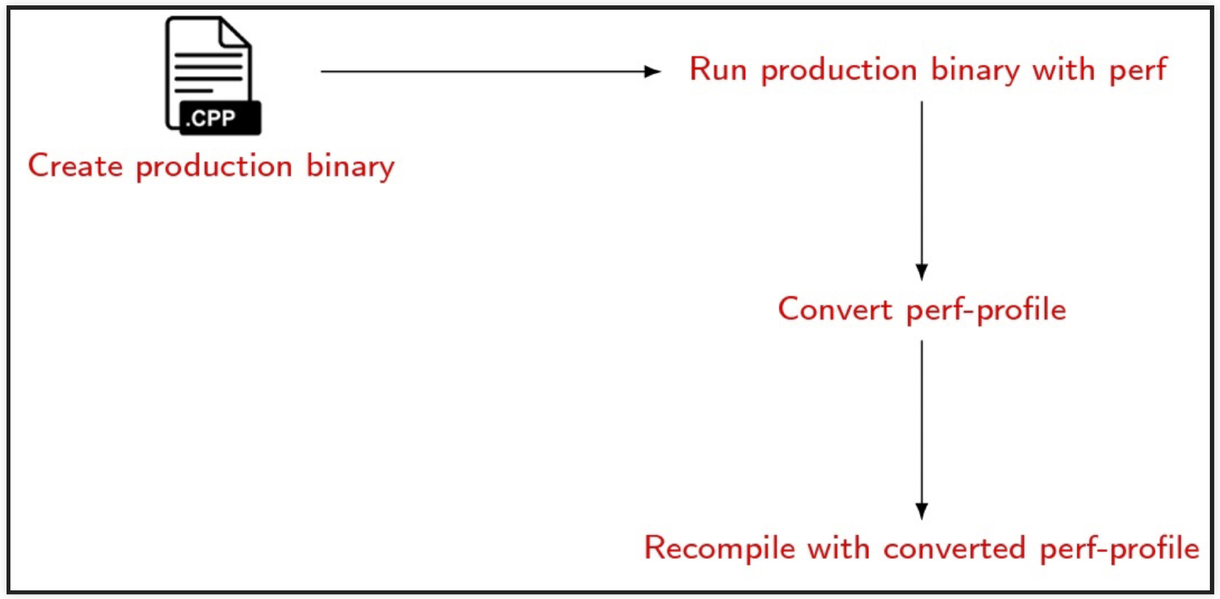
Goal: improve the computational performance of FullSim with compiler optimizations



- Use Profile Guided Optimization (PGO aka FDO) and Auto-FDO to improve performance
  - PGO/FDO: uses instrumented binaries. Results in potentially significantly slower instrumented binary
  - Auto-FDO: uses sampling hardware performance monitors
  - Both are aimed at optimizing code with lots of branching that is difficult to predict at compile time
- Will try both PGO and Auto-FDO
  - CMS implemented PGO and LTO with 10% performance gain
- Being investigated by Caterina Macron, Ben Morgan

AUTO-FDO STATUS

- Auto-FDO was attempted on simulation code in 2017
- Single static library did not exist: Auto-FDO could not modify library objects
- Will revisit Auto-FDO now that big static library is available



GEANT4 11 IN ATHENA

Goal: Test newer version (11.1) of Geant4 within Athena

- Initial success: building Geant4 11.1 in Athena and running short test job
- Updates to code in Athena needed when compiling against Geant4 11.1 caused no change in physics (when using Geant4 10.6)
- Currently checking which ATLAS patches applied on top of Geant4 10.6 are included in Geant4 11.1 already
- Needs more extensive Physics Validation once actually running with Geant4 11.1
- Being worked on by Ben Morgan and Mustafa Schmidt

FEEDBACK TO GEANT4: PHYSICS

- Close collaboration with Geant4 expert has resulted in new optimizations/fixes (EM physics, GammaGeneralProcess, Woodcock tracking)
  - Significant performance increases → about a factor of 2 improvement in event throughput
  - **We look forward to continuing this collaboration on projects such as G4HepEM and specialized tracking**
- **Change in pion energy resolution from Geant4 10.4 to 10.6 transition will have a xsnegative effect on jet energy resolution and Run 3 physics**
  - Resolving this issue is a high priority for ATLAS and the planned work by Geant4 on this is issue is very appreciated
- Neutron Fluence differences (~20% increase) between 10.1 and 10.6 observed behind thick materials (i.e., calorimeters)
  - Bugs were discovered and fixed but the difference remains significant.
  - Radiation studies use differences between FLUKA and Geant4 to estimate systematics for High-Luminosity LHC predictions
  - Will use a private build of Athena using G4 10.1 (instead of 10.6) for HL-LHC studies
  - **Accurate neutron physics in Geant4 is very relavant for the HL-LHC studies**

FEEDBACK TO GEANT4: CORE KERNELS AND R&D

- Core kernel
  - ATLAS acknowledges the effort of Geant4 to port ATLAS TileCal and HEC test beam code into the Geant4 validation portal.
  - **This achievement is very important because it allows studying the effects of Geant4 developments on ATLAS data and simulation**
- R&D
  - ATLAS is closely following developments concerning GPU workflows (e.g., Celeritas and AdePT) and is in close contact with the developers to test demonstrators that include pieces of ATLAS detectors
  - ATLAS recognizes the developments to make it easier to train fast simulation models.
    - However, fast simulation models are closely tied to detectors and require significant contact with and lead time for the experiments
    - To be effective, fast simulation developments must be relevant to real detector issues and have direct integration in the experiment's framework.

BACKUP