Studies of GEANT4 performance for different ATLAS detector geometries and code compilation methods

CATERINA MARCON

E. ELEN, JESSICA R. MADEIRA, B. MORGAN, O. SMIRNOVA, D. SMITH

ON BEHALF OF THE ATLAS COLLABORATION

VCHEP 2021

Outline

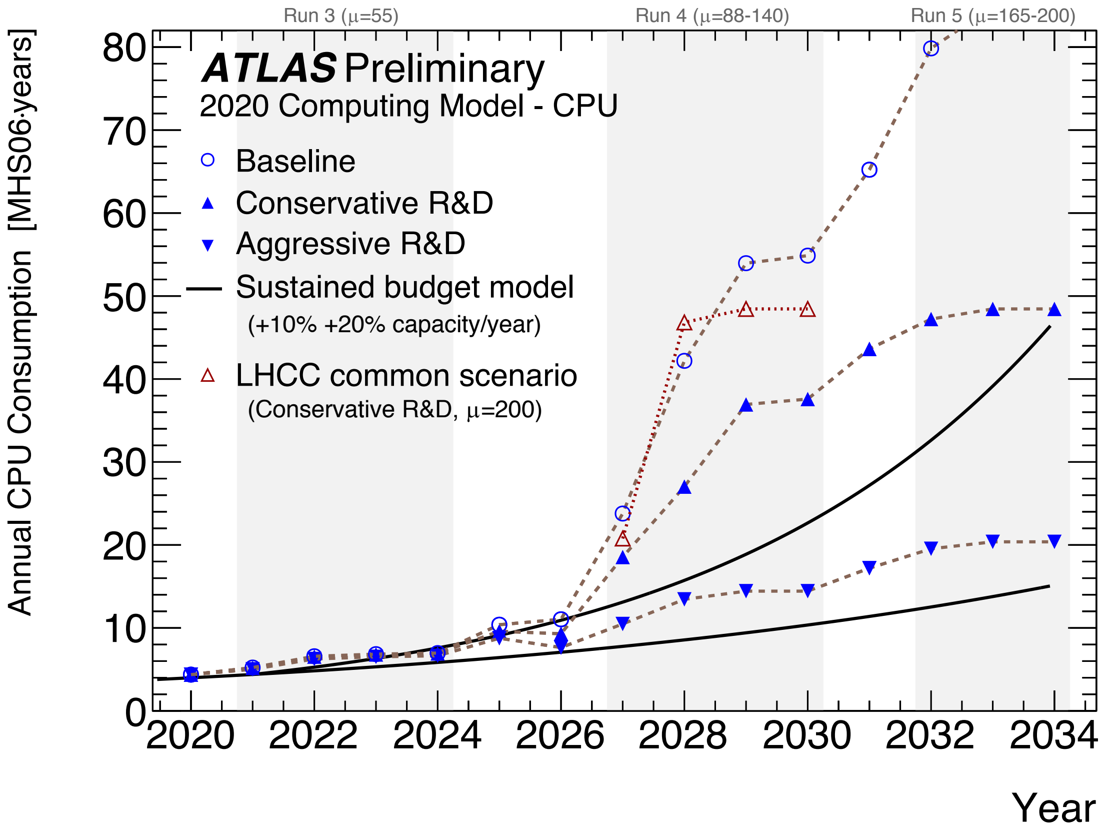
The aim of this study is to investigate different methods to reduce the full simulation execution time without sacrificing the quality of the simulated data and without altering the existing source code:

- A broad range of build-time configurations has been tested in order to check the independence of physics results from compiler-specific options;

- The impact on simulation execution time of different build types has been investigated;

- The effect of different primary particles on the simulation execution time has been studied.

Motivation



- During the High-Luminosity LHC phase, the goal is approximately ten times the data of early runs.
- This will require a corresponding increase in the amount of simulated data, which already accounts for 40% of ATLAS' CPU usage. Many analyses are already limited by the availability of MC events.
- An active R&D program to optimize the GEANT4 CPU requirements is a priority in ATLAS.

Method

- Run standalone Geant4 simulations, disentangling them from other libraries that may affect performance.
- Compiled G4 (version 10.5) both statically and dynamically. The computations have been carried out on a CERN standalone machine and on the Lund University cluster.
- Different versions of the GCC compiler, namely 6.2.0 and 8.2.0, Clang and ICC have been used for these investigations.
- As a benchmark, a Geant4 simulation (from A. Dotti [1]) has been used.
- The studies have been carried out using both a GDML[1]-based ATLAS geometry (without EMEC[2]) as well as the GeoModel representation of the full ATLAS geometry.

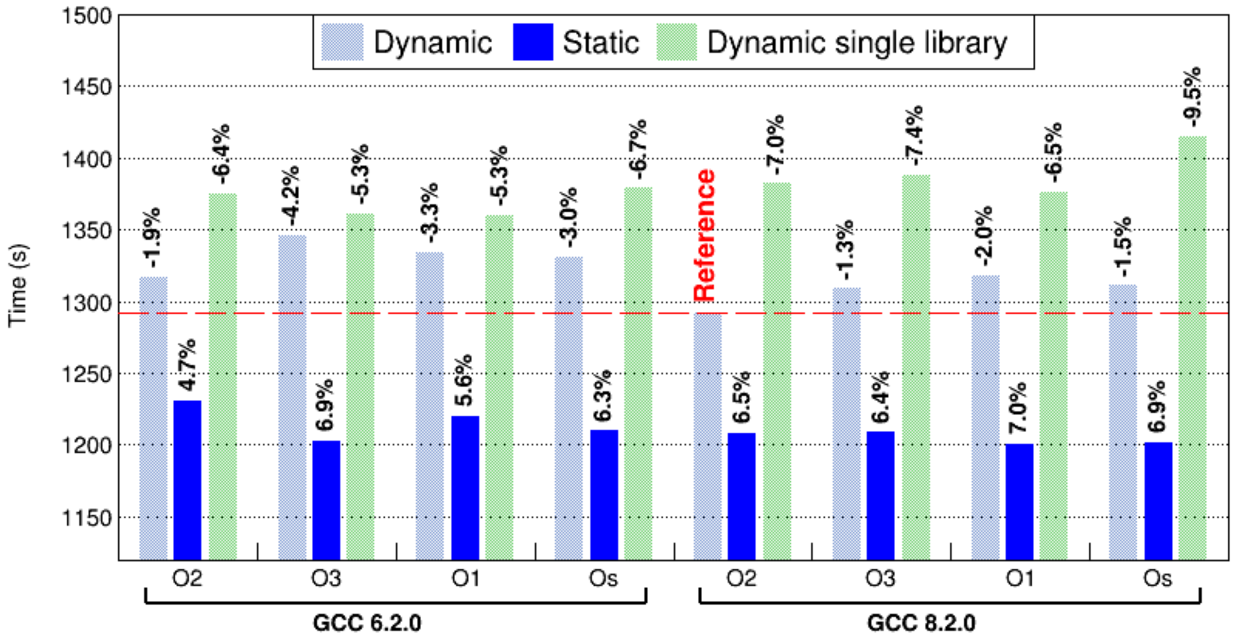
[1] https://gitlab.cern.ch/adotti/Geant4HepExpMTBenchmark

1 Geometry Description Markup Language: an XML-based, application-independent geometry description format

2 ElectroMagnetic EndCap calorimeter

LUND UNIVERSITY

Single dynamic library



- Three different build types, default dynamic multi-library, dynamic single library and static library, have been used.
- The study is performed using 2500 initial events and 4 threads. The GDML ATLAS geometry has been considered.
- To build the single dynamic GEANT4 library the CMake structure has been modified adding the new flag BUILD_SINGLE_LIB.
- The dynamic single library approach, for both GCC versions, increases the execution time up to 10%.
- This effect could be explained considering the trampoline used by dynamic libraries to provide the calling functions with the memory addresses of the called methods.

Impact of different primary particles using GDML geometry

Particle type	Simulation time (s)	Decrease w.r.t. protons (%)	Increase w.r.t. static case (%)
Dynamic library (10 GeV, 20 runs)			
p	601 ± 9	—	9.9
π^-	594 ± 10	1.1	10.4
π^+	577 ± 5	4.2	9.6
Geantino	3.0 ± 0.1	1.99×10^4	5.6
Static library (10 GeV, 20 runs)			
p	546 ± 6	—	—
π^-	538 ± 8	1.5	—
π^+	526 ± 4	3.9	—
Geantino	3.2 ± 0.1	1.7×10^4	—
Dynamic library (20 GeV, 100 runs)			
p	1130 ± 14	—	10.5
π^-	1083 ± 19	4.3	9.4
π^+	1079 ± 18	4.7	9.7
Geantino	3.1 ± 0.1	3.64×10^4	4.7
Static library (20 GeV, 100 runs)			
p	1023 ± 12	—	—
π^-	990 ± 14	3.3	—
π^+	983 ± 12	4	—
Geantino	3.0 ± 0.1	3.4×10^4	—

- Several primary particles (protons and charged pions) of different energies have been considered;
- For all the primary particles analyzed, a decrease in the simulation execution time is observed for the static build;
- This improvement is increasingly pronounced as the complexity of the interactions grows: the speed-up exceeds 10% in case of 20 GeV protons tested with this geometry.

LUND UNIVERSITY

Impact of different primary particles using full ATLAS geometry

Particle	Simulation time (s)	Decrease w.r.t. protons (%)	Increase w.r.t. static case (%)
Dynamic library (10 GeV, 45 runs)			
p	647 ± 9	—	6.2
π^-	657 ± 9	−1.5	6.3
π^+	640 ± 10	1.1	7.0
Static library (10 GeV, 45 runs)			
p	609 ± 10	—	—
π^-	618 ± 9	−1.5	—
π^+	598 ± 7	1.8	—
Dynamic library (20 GeV, 45 runs)			
p	1212 ± 13	—	6.5
π^-	1209 ± 18	0.2	6.7
π^+	1181 ± 14	2.6	6.1
Static library (20 GeV, 45 runs)			
p	1138 ± 11	—	—
π^-	1133 ± 14	0.4	—
π^+	1113 ± 13	2.2	—
Dynamic library (50 GeV, 45 runs)			
p	2797 ± 46	—	6.1
π^-	2752 ± 40	1.6	6.9
π^+	2715 ± 39	3.0	6.9
Static library (50 GeV, 45 runs)			
p	2636 ± 30	—	—
π^-	2573 ± 28	2.4	—
π^+	2539 ± 37	3.8	—

- Also with the full ATLAS geometry, a substantial decrease in the simulation execution time is observed for the static build even though the inclusion of the EMEC reduces by ~3% the gain of the static build.

- With both the geometries:
  - the static build shows a tendency to be less sensitive to the type of primary particles used;
  - for protons with 20 and 50 GeV, computations show a longer execution time: the proton undergoes more ionization processes in the medium it traverses and travels longer distances before absorption;
  - fluctuations between the pions are consistent with the slightly larger interaction cross section of the negative particle at the considered energy.

LUND UNIVERSITY

Conclusions & outlook

- Several factors can significantly affect the full GEANT4 simulations execution times:
  - Tests with the single dynamic library resulted in a $\sim$ 10% increase in the execution time; the effect is ascribed to the trampoline/lookup table mechanism of dynamic linking.
  - All the investigations, performed with different primary particles, carried out with the static build type with the GDML geometry (without EMEC), have shown a gain of about 10% with respect to the reference multi-library dynamic case.
  - Evaluations of the GeoModel geometry, representing instead the full ATLAS detector (including EMEC), confirm a gain with the static build type even if smaller (about 7%).
  - Unsafe math optimizations as well as certain compilers, namely the Clang family and older GCC versions, may have a negative impact on the quality of the physics results (e.g. a shift in the average energy deposition).
- As a future step, studies with static build could be integrated and tested in the environment of the Athena framework.

LUND UNIVERSITY

LUND
UNIVERSITY

Overview of computational resources

	CERN standalone machine	Compute node on Lund University cluster
CPU	2× Intel Xeon E5-2630 v3 2.40GHz	2× Intel Xeon E5-2650 v3 2.30GHz
Architecture	64 bit Haswell x86_64	64 bit Haswell x86_64
N. of cores	16	20
Threads per core	2	1
Cache	20 MB (L1: 64 KB, L2: 256 KB, L3: 20 MB)	25 MB (L1: 64 KB, L2: 256 KB, L3: 25 MB)
RAM	64 GB	128 GB
Filesystem	XFS	IBM General Parallel File System (GPFS)
Operating system	CentOS 7	CentOS 7

LUND UNIVERSITY