Fermilab U.S. DEPARTMENT OF ENERGY Office of Science

CMS G4 Simulation Application: Computing Performance

V. Daniel Elvira, Vladimir Ivantchenko, Kevin Pedro

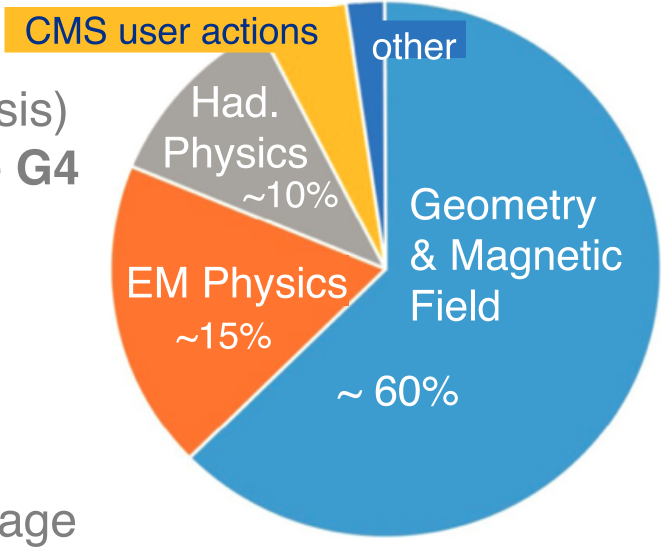
Geant4 Collaboration Meeting

Lund, Sweden, August 27th, 2018

(basically K. Pedro's CHEP talk)

Overview

- CMS full simulation uses **Geant4**
- **Run 1, 2:** MC production (gen, sim, reco, analysis) took 85% of total CMS CPU resources, with the **G4 module taking 40% of the total**
- Largest contributors to CPU usage in Geant4: geometry, magnetic field, EM physics
- CMS has implemented numerous technical options and approximations to improve CPU usage in full simulation application (fast simulation techniques – using ATLAS language)
- Continue to explore new options and improvements
  - Including the GeantV transport engine, effective use of HPC, ML for fast simulation

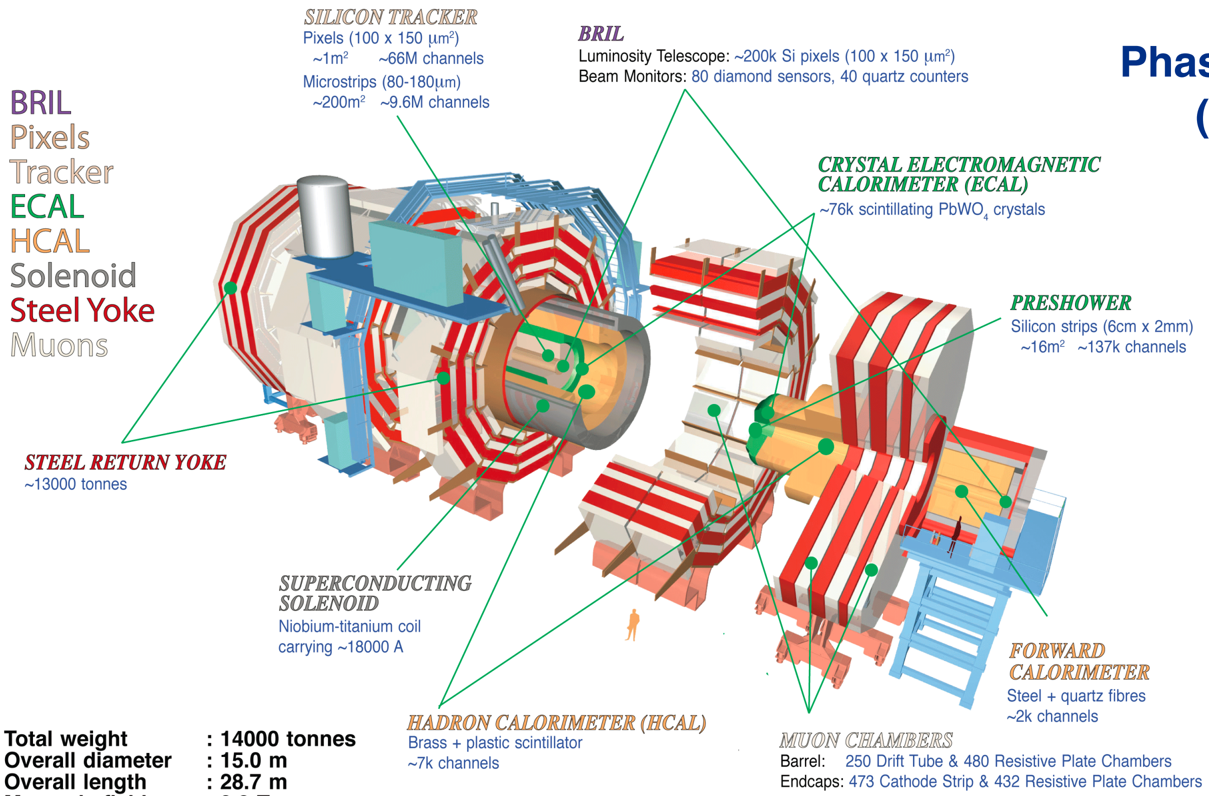


Geant4 10.0p02
(similar in other versions)

Fermilab

The CMS Detector

Phase 0 Detector (Run 1, 2)



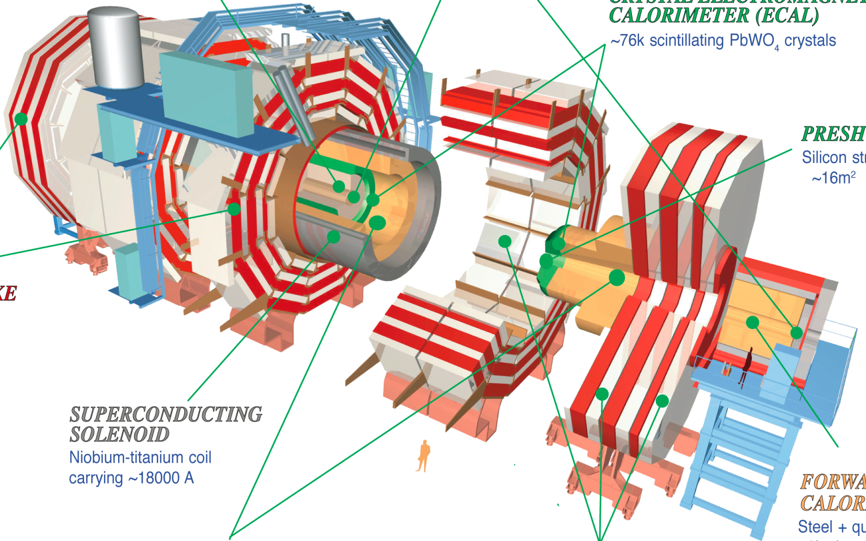
Total weight	**: 14000 tonnes**
Overall diameter	**: 15.0 m**
Overall length	**: 28.7 m**
Magnetic field	**: 3.8 T**

The CMS Detector

Phase 1 upgrade (Run 3)

127M ←

BRIL
Pixels
Tracker
ECAL
HCAL
Solenoid
Steel Yoke
Muons



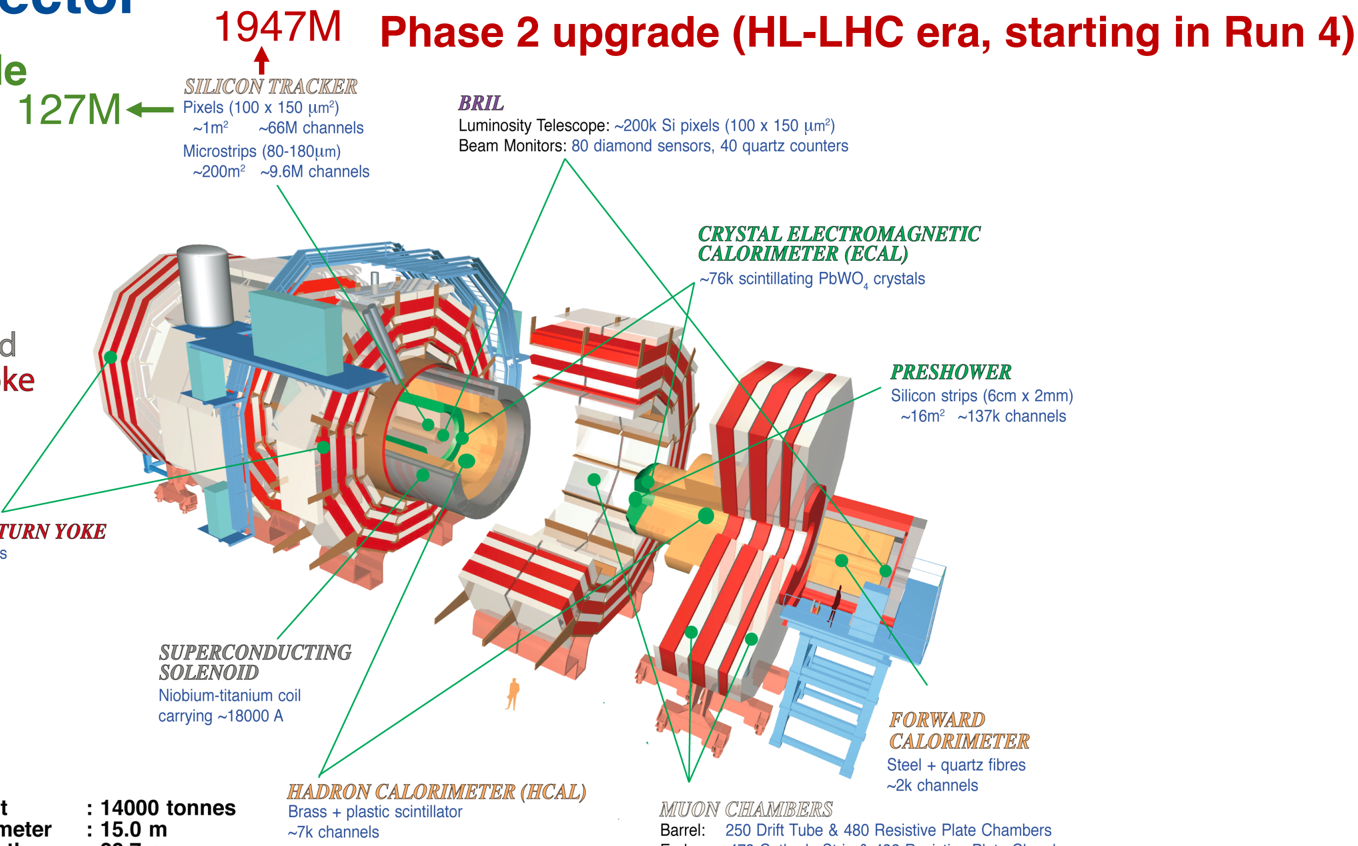
Total weight	**: 14000 tonnes**
Overall diameter	**: 15.0 m**
Overall length	**: 28.7 m**
Magnetic field	**: 3.8 T**

The CMS Detector

Phase 1 upgrade (Run 3)

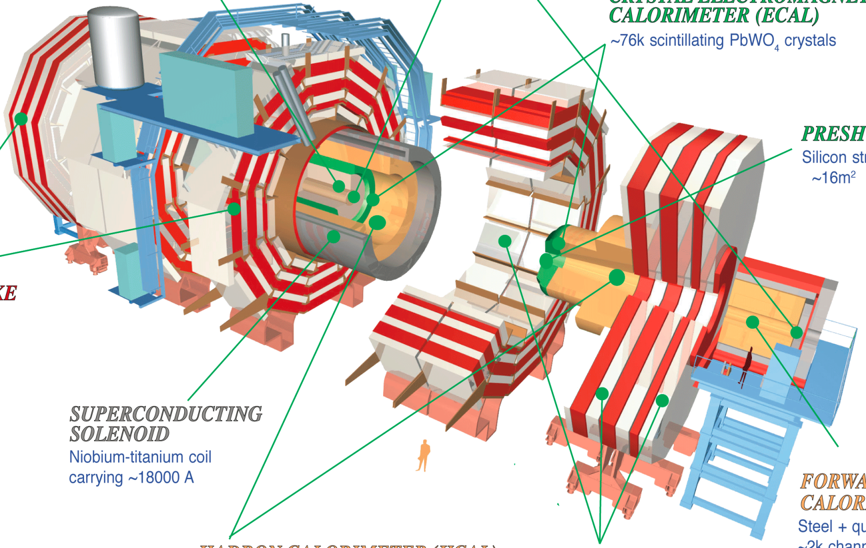
Phase 2 upgrade (HL-LHC era, starting in Run 4)

BRIL
Pixels
Tracker
ECAL
HCAL
Solenoid
Steel Yoke
Muons



Total weight	**: 14000 tonnes**
Overall diameter	**: 15.0 m**
Overall length	**: 28.7 m**
Magnetic field	**: 3.8 T**

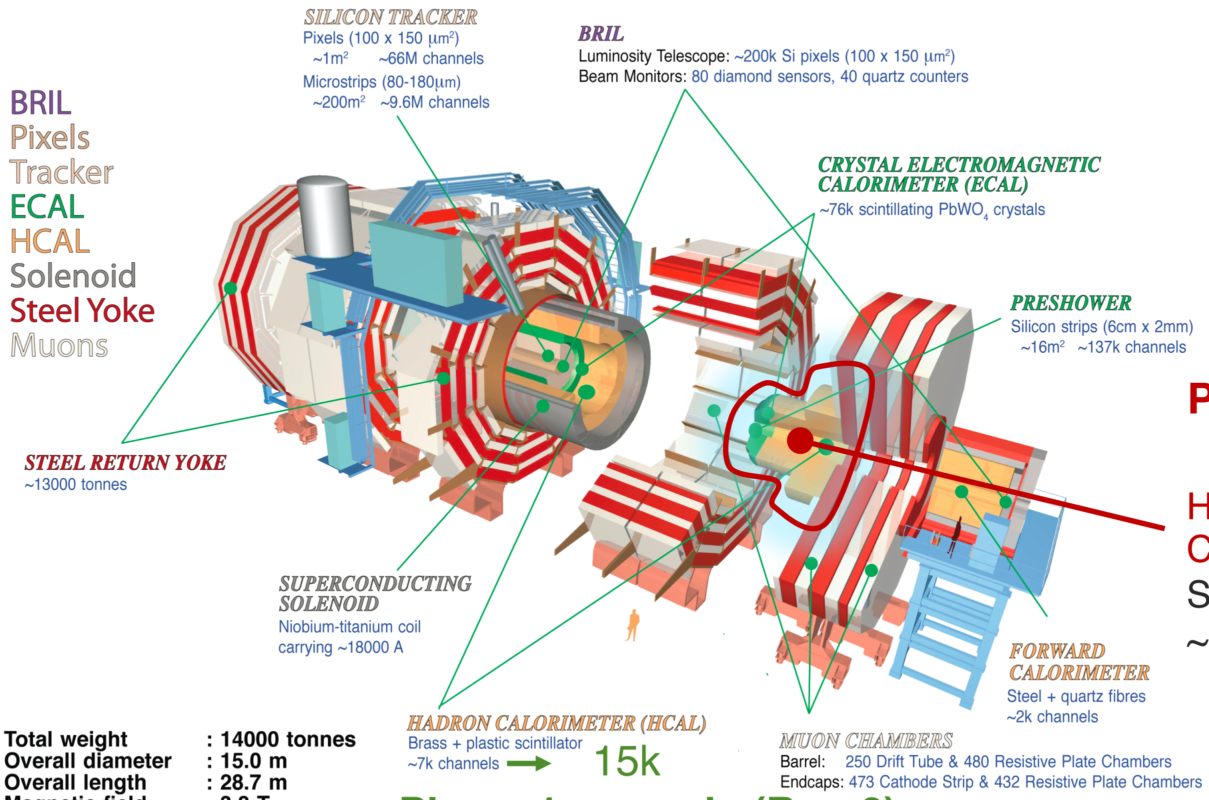
The CMS Detector



Total weight	**: 14000 tonnes**
Overall diameter	**: 15.0 m**
Overall length	**: 28.7 m**
Magnetic field	**: 3.8 T**

Phase 1 upgrade (Run 3)

The CMS Detector



Phase 2 upgrade (HL-LHC)

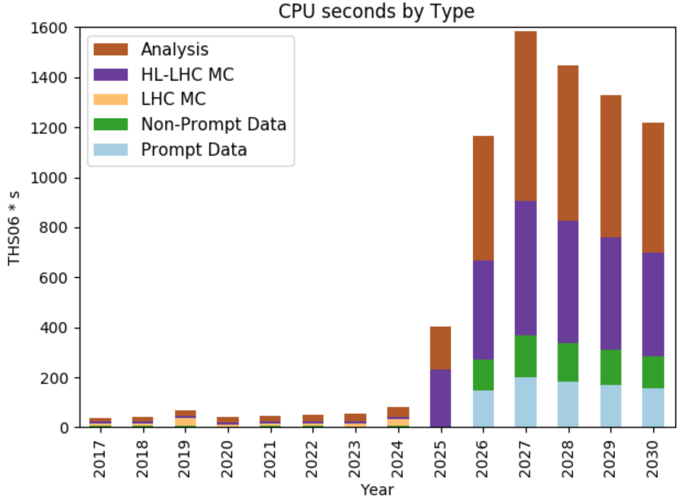
High Granularity Calorimeter (HGCal)
Silicon, scintillator
~6M channels

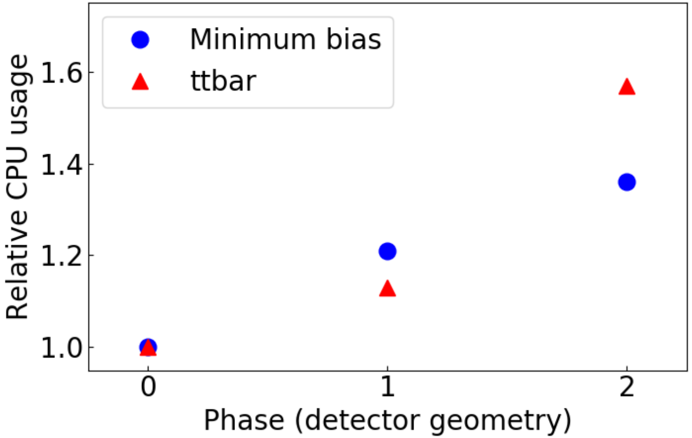
Total weight	**: 14000 tonnes**
Overall diameter	**: 15.0 m**
Overall length	**: 28.7 m**
Magnetic field	**: 3.8 T**

Phase 1 upgrade (Run 3)

Challenges of HL-LHC Era

- CMS Phase 0 and Phase 1 simulation geometries have 2.1 million elements
- Phase 2 geometry has 21.9 million elements
  - Increase in CPU time for simulation





- Simulate more events to keep up with HL-LHC data volumes: 10×(Phase1)
- May also need to switch to more “expensive” physics lists to simulate HGCal

Reconstruction will take longer (and a larger fraction of CPU resources) due to pileup, granular detectors

But need more simulated events, better physics accuracy, in a more complex geometry

Existing Improvements as of end of 2017

- **Static library**: avoid calls to procedure linkage table (PLT) for dynamic loading of libraries
- **Production cuts**: 0.01mm (pixel), 0.1mm (strip tracker), 1 mm (ECAL/HCAL), 0.002 mm (muon systems), 1 cm (support structure)
- **Tracking cut**: 2 MeV (within beampipe) → avoid looping electrons
- **Time cut**: 500 ns
- **Shower library**: use pre-generated showers in forward region (HF, ZDC, Castor)
- **Russian roulette**: discard N-1 neutrons < 10 MeV or gammas < 5 MeV (in calorimeters), retain N^{th} particle and assign it a weight of N
- **FTFP_BERT_EMM**: modified physics list, simplified multiple scattering model for most regions (default used for HCAL, HGCal)
  - When all optimizations applied together, CMS achieves ~3–5× speedup!

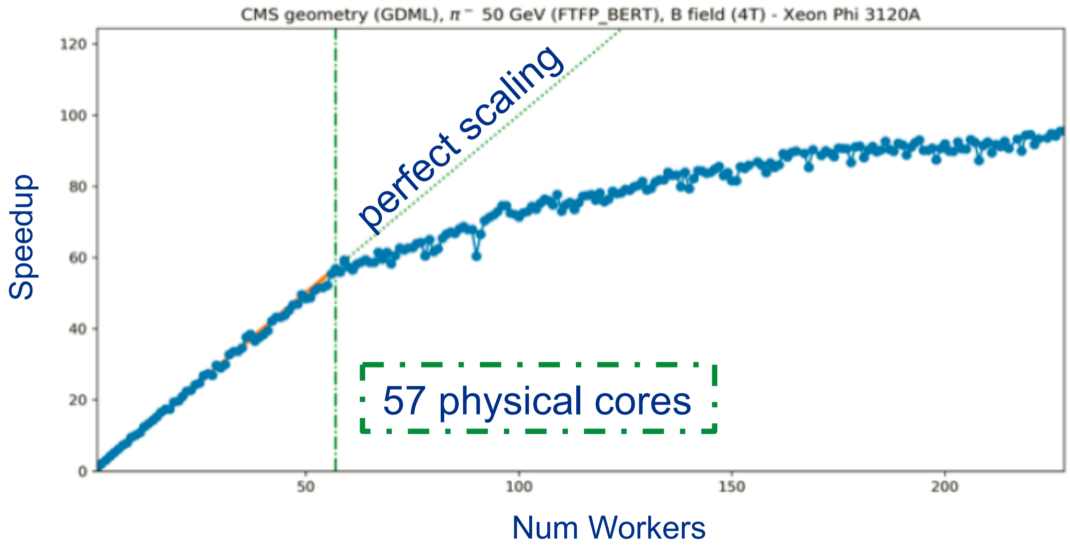
Time performance impact of existing improvements

Configuration	Relative CPU usage: MinBias	Relative CPU usage: ttbar
No optimizations	1.00	1.00
Static library	0.95	0.93
Production cuts	0.93	0.97
Tracking cut	0.69	0.88
Time cut	0.95	0.97
Shower library	0.60	0.74
Russian roulette	0.75	0.71
FTFP_BERT_EMM	0.87	0.83
All optimizations	0.21	0.29

- From HEP Software Foundation Community White Paper
  - CMS Phase 0 detector, Geant4 10.2
- HF shower library, Russian Roulette have largest impacts
- Cumulative effects: with all improvements, simulation is 4.7× (3.4×) faster for MinBias (ttbar)
- CMS simulation takes 4.3 sec†/event (24.6 sec†/event) for MinBias (ttbar)

†1 sec = 11 HS06 for test machine

Multithreading



CMSSW multithread capable as of:

- Run 2 (Spring 2016) for reco
- Run 2 (Spring 2017) for generator and Geant4 modules

- Geant4 includes event-level multithreading
- Nearly perfect scaling with physical cores, further 30% gain from hyperthreading
- Memory reduced by factor of 10 (vs. multiprocessing approach)
- Similar gains in throughput observed, memory usage remains under 2GB

➢ More efficient use of grid resources (included in CMS production releases)

2018 improvements: geometry

VecGeom: new library for detector geometry

- Supports vectorization and new architectures
- Code rewritten to be more modern and efficient (vs. Geant4, ROOT, USolids)
- Can be used in scalar mode with Geant4
- CMS observes 7–13% speedup with similar memory usage

  → Just from code improvements, no vectorization!

➢ Included in latest CMS production releases to be used in 2018 physics program

  ○ First mainstream use of vectorized library by experiment

	Relative CPU usage	
Geometry library	**MinBias**	**ttbar**
Native	1.00	1.00
VecGeom	0.87	0.93

Fermilab

2018 improvements: magnetic field

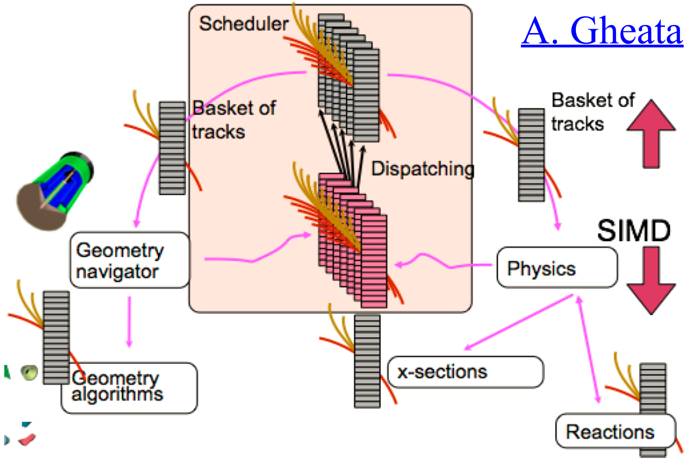
- Faster stepper (G4DormandPrince745) for tracking in magnetic field
  - Also a more robust algorithm
- Smart tracking: energy-dependent propagation through EM fields
- CMS observes 8–10% speedup with these optimizations (preliminary)
  - Enabled by migration to latest Geant4 version 10.4

	Relative CPU usage	
Stepper	**MinBias**	**ttbar**
G4ClassicalRK4	1.00	1.00
G4DormandPrince745	0.93	0.98
G4DormandPrince745 + smart tracking	0.92	0.90

(tested w/ gcc 7.0 and 16 threads)

Potential improvements: the GeantV transport engine

- CMS has already achieved significant speedups in Geant4 and enabled event-level multithreading for more efficient use of resources
- However, even this will not suffice for the demands of the HL-LHC era
- Enter GeantV: Vectorized Transport Engine
  - Track-level parallelism: process multiple events simultaneously
  - Exploit single instruction, multiple data (SIMD) vectorization
  - Group similar tracks into *basket* (based on particle type, geometry/material)
  - Send entire basket to algorithm: process particles in parallel

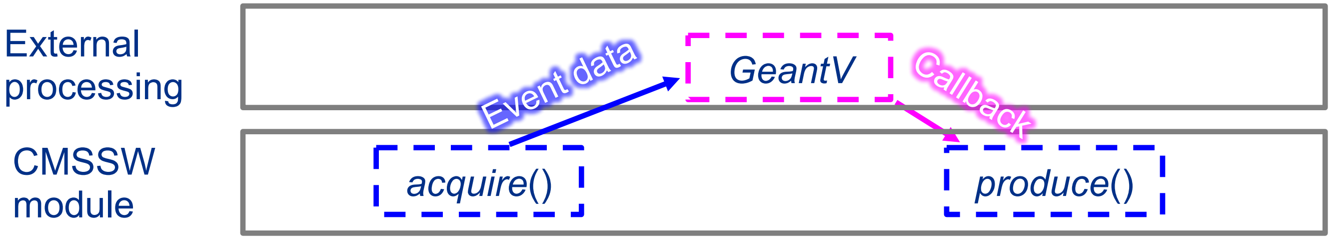


A. Gheata

Fermilab

Early Testing of GeantV in CMSSW

- Started with integration into toy-mt-framework → included in alpha release
  - Used for CMS multithreading R&D (Intel Thread Building Blocks)
- ➢ Now have a working example compatible w/ CMSSW development release
- Run GeantV in "external loop" mode using CMSSW ExternalWork feature:
  - Asynchronous task-based processing



- ➢ Co-development approach: test consistency of threading models, interfaces
  - Provide feedback to prevent divergence between CMS and GeantV
    - ✓ This is the main reason why early testing and co-development is ESSENTIAL

Elements of GeantV integration

The GeantV interfaces will change if/when integrated to Geant4. As of now:

- ✓ Generate events in CMSSW framework, convert HepMC to GeantV format
- ✓ Build CMSSW geometry natively and pass to GeantV engine
- Using constant magnetic field, limited EM-only physics list
- Sensitive detectors and scoring not yet adapted to new interfaces
- Production cuts also not yet included
- ➢ First integration of GeantV into experimental software framework
  - Run with elements specified above
  - Integration with downstream steps (e.g. digitization): longer timescale, requires more development for thread-safe scoring
- CMS will test GeantV beta release, targeting demonstration of speedup
  - Community decision to support GeantV engine as part of Geant4 on timescale of HL-LHC

Summary

- CMS has substantially reduced CPU usage of Geant4 full simulation
  - ~3–5× speedup using various technical improvements and physics-preserving approximations
  - Continue to find improvements on the order of 10%, e.g. from VecGeom and magnetic field stepper/tracking optimizations
- HL-LHC and Phase 2 upgrades bring significant challenges:
  - ➢ Need more simulated events, more physics accuracy, in more complicated geometry… although G4 simulation will take a smaller fraction of total CPU usage
- GeantV transport is one promising approach to speed up full simulation even further
  - Alpha release is available, beta release planned for 2019
  - ➢ Successful early integration in CMS software framework!
    - Aim for 2–5× speedup with final version
- Efficient use of HPC systems may be a new requirement from funding agencies
- ML techniques will be explored for fast simulation - will probably not replace full sim

CMS simulation: next 4+ years

Full simulation only schedule – CMS also has program to use Fast Sim techniques in its Geant4 full simulation application (machine learning, parametrizations), and to execute full simulation in a hybrid cloud environment which includes HPC systems. (The GeantV transport engine design utilizes SIMD vectorization for fine-grained parallelism.)

R&D timeline: (targets HL-LHC)

- GeantV alpha tag within Toy/actual CMSSW: tests of multithread models, track-level basketization, user interfaces
- Development of Computing Performance (CP) infrastructure for Geant4 versus GeantV CP tests within CMSSW
- GeantV beta tag within CMSSW: CP tests to evaluate speedup, iteration with developers
- Community decision. If YES, iteration with developers: features, speed, integration to Geant4 toolkit
- Commissioning 202X-2025/2026: computing performance, robustness, physics validation: full support from Geant4 Collaboration

?

?

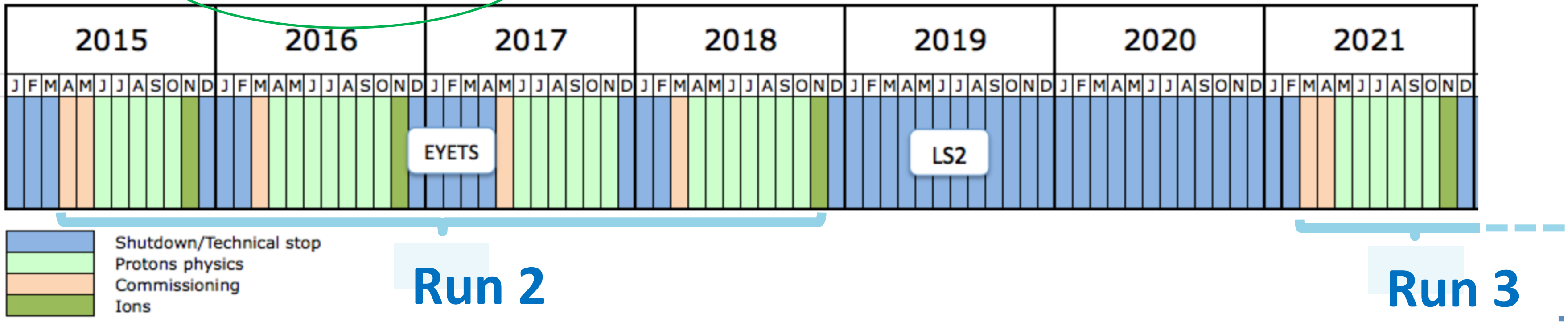
?

Production timeline:

Done!

Upgraded shower libraries, G4 DormandPrince stepper, smart tracks, RN handling for full reproducibility

DD4HEP migration, improved pileup infrastructure, potential G4 upgrades (new releases, improved navigator, predictions with systematic uncertainties – VMP, other modular packages from R&D)



Fermilab