Computing Performance of CMS Detector Simulation in HL-LHC

Soon Yung Jun, Sunanda Banergee, Daniel Elvira (Fermilab)
Vladimir Ivantchenko (CERN)

HSF-Simulation Meeting

Feb, 2017

Outline

- Status of CMS simulation
- Plans for simulation benchmarking
- Recent results of standalone benchmark for Geant4 10.3

- Today we mainly focus on full simulation and Geant4 CPU/memory benchmarking

- Recent CMS talks includes all aspects of CMS simulation:
  - *Sezen Sekmen*, "Recent Developments in CMS Fast Simulation", ICHEP'16 proceedings, arXiv:1701.03850
  - *Sunanda Banerjee* "Validation of Physics Models of Geant4 using data from CMS Experiment" CHEP'16 #334
  - *David Lange* "Upgrades for the CMS simulation" CHEP'16 #361

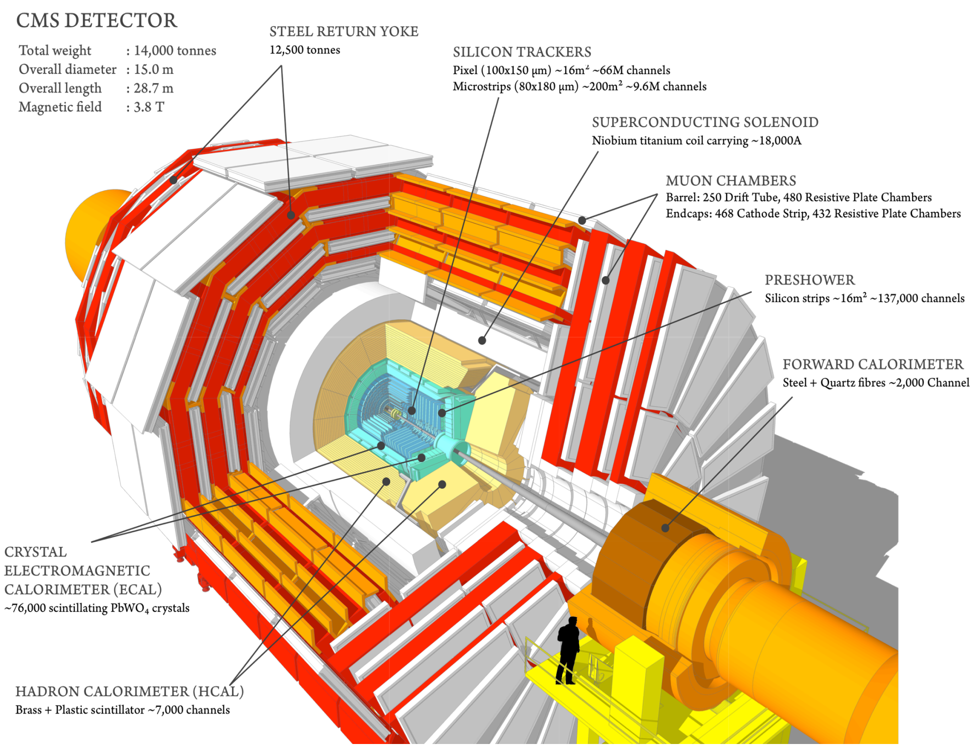
CMS simulation faces significant challenges for both today and tomorrow

Higher LHC luminosity means:

- Need for more accurate simulations
- Need for more events (ideally more events/CHF)
- More demanding pileup simulation requirements

Major detector upgrades in 2017, during LS-2 and for HL-LHC mean:

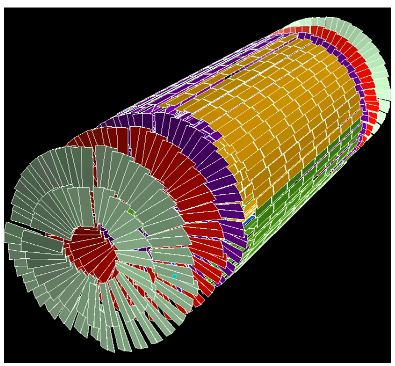
- New detector concepts to develop, benchmark and validate
- The need to make reliable simulations for HL-LHC luminosities

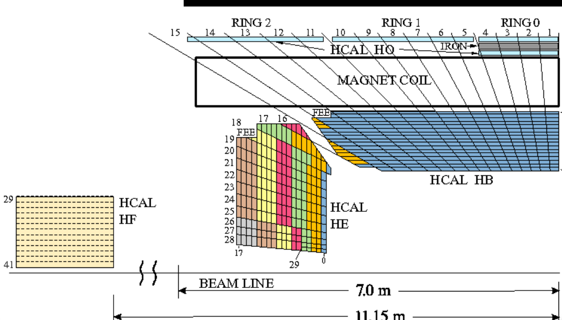


Increases in LHC luminosity and CMS detector upgrades push our simulation capabilities

Detector upgrades push continued simulation development in CMS

2017 upgrades



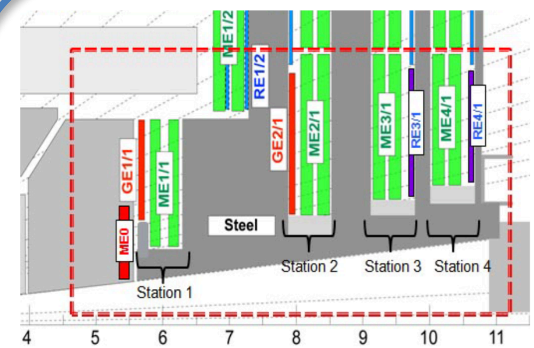


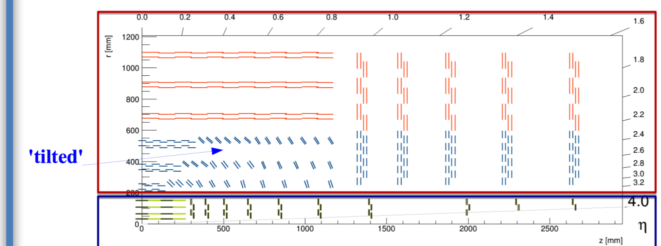
Muon ID

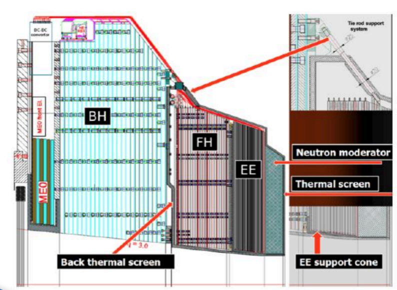
Tracking

Calorimetry

HL-LHC upgrades



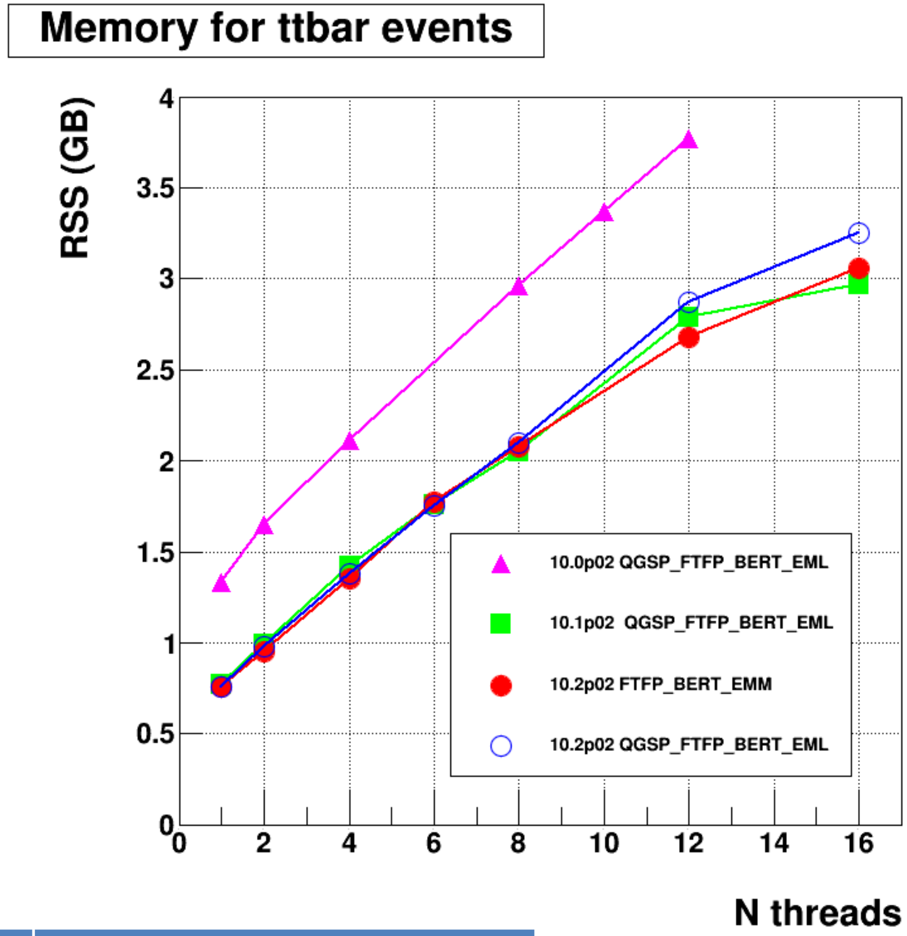




Status of CMS full simulation

- Run-2 simulation production
  - Geant4 10.0p02 + CMS patches in sequential mode
  - Production platform slc6_amd64_gcc530
  - Physics List QGSP_FTFP_BERT_EML
  - 16 billion events produced
- 2017 simulation production
  - Geant4 10.2p02 + CMS patches in MT mode
  - Production platform slc6_amd64_gcc620
  - Physics List FTFP_BERT_EMM
  - Massive production not yet started

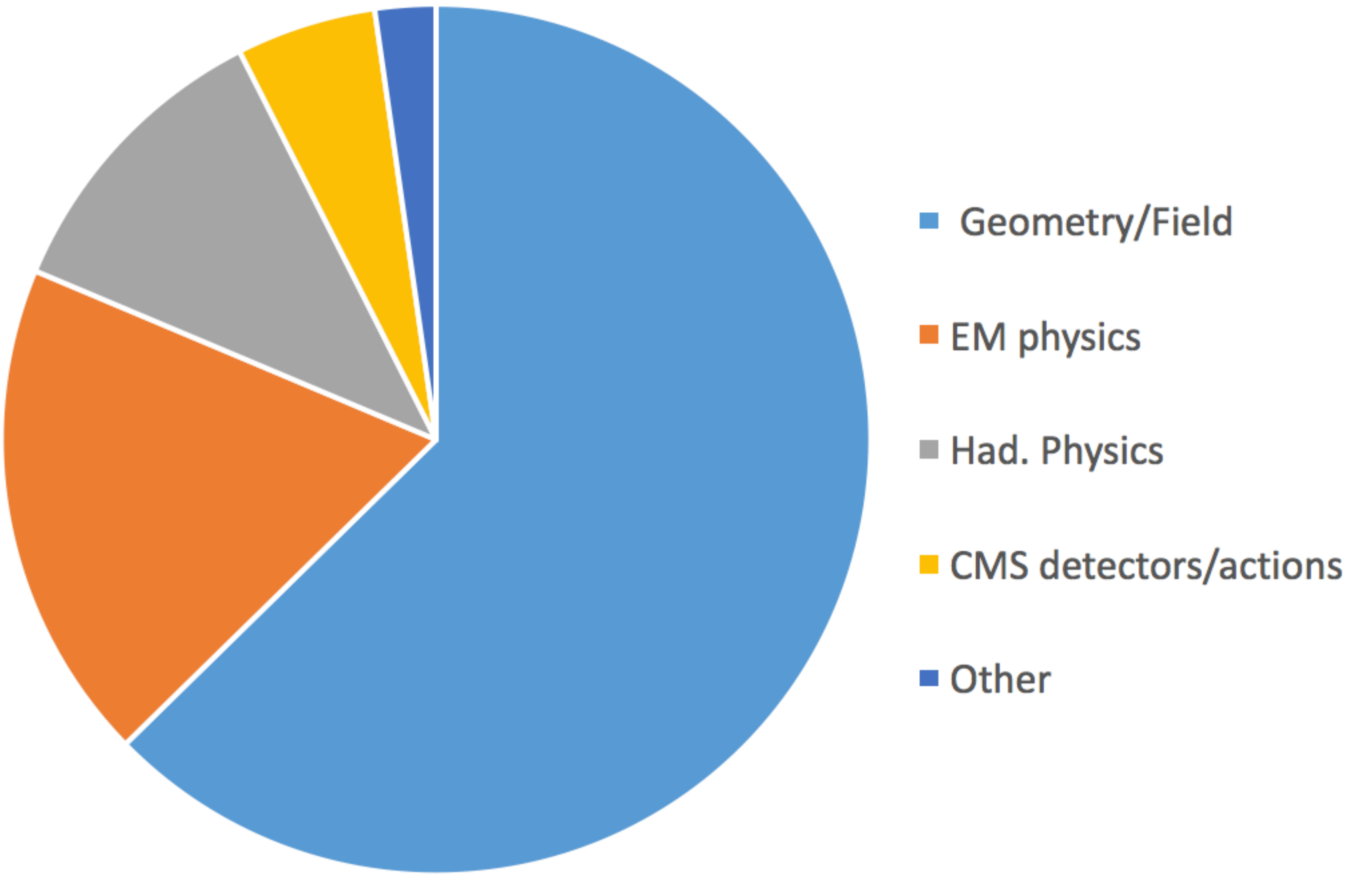
Memory for CMS run with Geant4 10.2p02



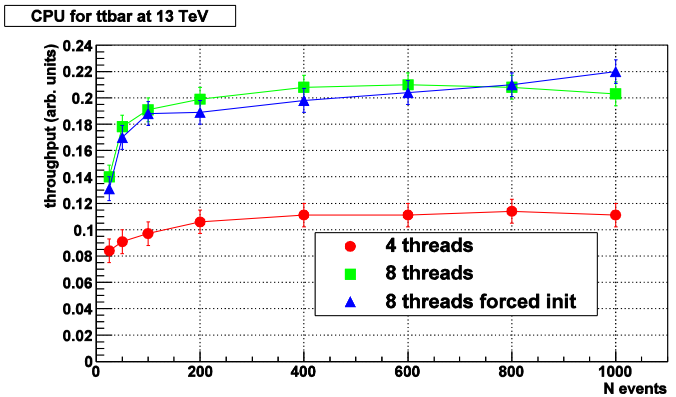
- A node with 12 Intel cores was used to study memory utilisation
- 13 TeV hard scattering event were simulated
  - Results after 1000 events are shown
  - CMS private patches to 10.0 include backports of fixes of memory leak and memory optimisation
  - Results for 10.1 and 10.2 are practically the same
  - No dependency on Physics List
- No problems to run CMS SIM production in the MT mode

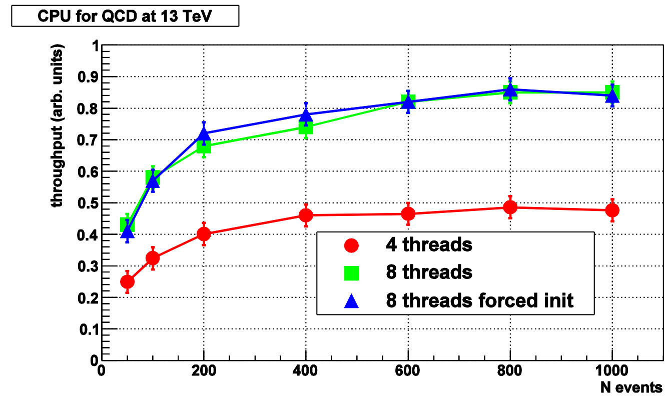
Release	1st thread (GB)	Delta per thread (GB)
10.0p02+CMS patches	1.33	0.23
10.2p02	0.76	0.19

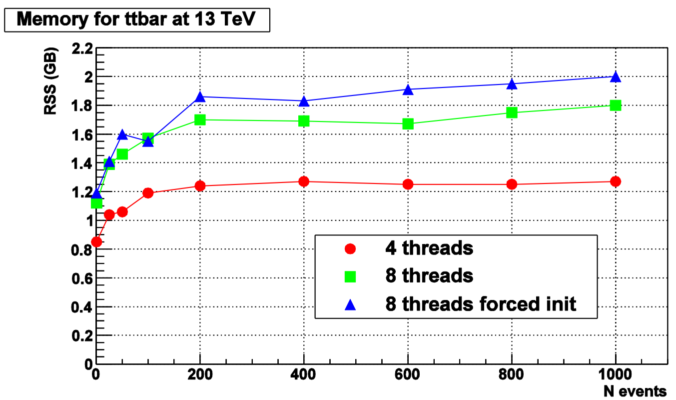
Where our simulation CPU goes

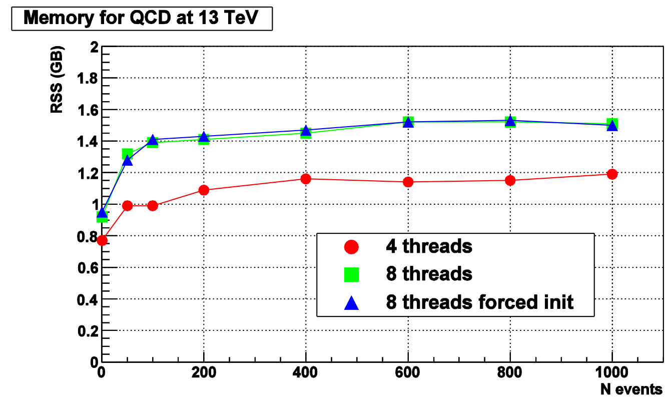


Dynamic of CPU and RSS for 13 TeV CMS simulation run in MT mode for Geant4 10.2p02









- Maxmum CPU efficiency is achieved after simulation ~500 events
  - The turnon shape is expected due to
    1. CMS and Geant4 initialisation before 1st event
    2. Lazy initialization of hadron physics in Geant4

CMS CPU performance for 2017

Geant4 version and configuration	Number of threads	CPU for ttbar (events/thread relative rate)	CPU for QCD (events/thread relative rate)
10.0p02 seq.	1	1	6.0
10.2p02 MT	1	1.07	5.7
10.2p02 MT	4	1.06	5.1
10.2p02 MT	8	1.05	5.0

CMS 2015/6

CMS 2017

- Overall, no substantial changes in our technical performance since 2015
  - Modest improvements in CPU performance in Geant4 10.2p2 and in CMS sensitive detector code offset by other changes
- No noticible performance loss (or gain) with threading (up to 8 threads)

CMS plans for simulation benchmarking

- CMS detector simulation at HL-LHC
  - New vertex and tracker detectors, upgraded calorimeters and muons systems:
    - New requirements for CPU/memory/accuracy
    - Potential use of a refined set of physics list
    - Options for adopting new development (VecGeom, …)
      - More details in Daniel Elvira talk, “CMS Simulation in the HL-LHC era” at the HSF-CWP workshop
- Perspectives of computing performance of CMSSW in the HL-LHC era
  - Preliminary performance profiling with a standalone CMS simulation and Geant4 (already available)
  - Performance profiling with CMSSW for new Geant4 versions
  - Regular performance monitoring and benchmarking as new features are integrated

First results of standalone performance monitoring

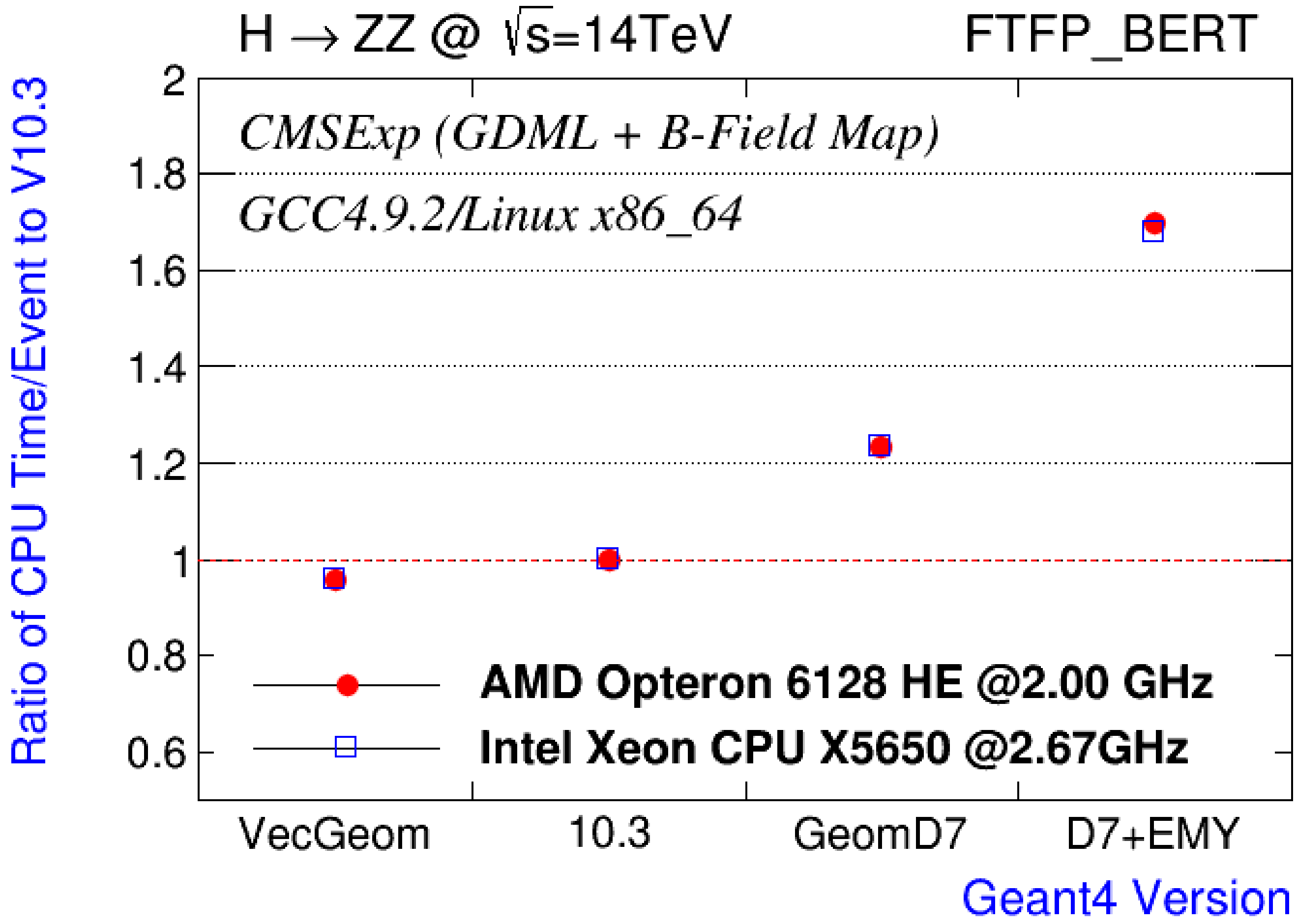
Soon Yung Jun

Preliminary Computing Performance with Geant4 10.3

- cmsExp: a standalone Geant4 application with
  - cms.gdml (originally converted from cms.root in 2012)
  - a custom B-Field map excerpted from CMSSW (5.1.0): a volume-based interpolated B-field based on1cm-grid of B(r,z) (900x1600 points - also an option to use parametrized B-field inside the CMS tracker)
- Benchmark
  - Reference: Geant4 10.3 with FTFP_BERT
  - VecGeom: v00.03.00 with Geant4 10.3
  - GeomD7: CMS upgraded tracker geometry (D7)
  - EMY: GeomD7 + FTFP_BERT_EMY physics list
- Profiling Details
  - 50 events of $H \rightarrow ZZ$ produced by PYTHIA @ $\sqrt{s} = 14$ TeV
  - Average information over 96x2 jobs on AMD + 36 jobs on Intel

cmsExp Computing Performance with Geant4 10.3

- CPU: Performance is normalized to 10.3



(detail profiling information at https://g4cpt.fnal.gov/)

cmsExp Performance with Geant4 10.3 + VecGeom

- Source of 5% CPU gain is not clear as there are many stuck tracks especially from PolyCone

Default: (top consumer by G4Solid)

G4PolyconeSide::DistanceAway	(1.52%)
G4PolyconeSide::Distance	(0.86%)
G4PolyconeSide::Inside	(0.55%)
G4IntersectingCone::LineHitsCone1	(0.54%)
G4PolyconeSide::PointOnCone	(0.52%)
G4PolyconeSide::Intersect	(0.41%)

VecGeom: (top consumer form vecgeom)

vecgeom::cxx::USolidsAdapter::DistanceToOut	(0.35%)
G4USolid::Inside	(0.32%)
vecgeom::cxx::ScalarShapeImplementationHelper	(0.26%)
vecgeom::cxx::USolidsAdapter::DistanceToIn	(0.20%)
vecgeom::cxx::USolidsAdapter::Inside	(0.19%)
vecgeom::cxx::USolidsAdapter::DistanceToOut	(0.18%)

- Need a stable version of VecGeom without errors from Geometry

cmsExp Performance with Geant4 10.3 + GeomD7 + EMY

- Summary of CPU degradation

20% CPU degradation from 10.3 to GeomD7:

Top affender
G4CrossSectionDataStore::GetCrossSection (2.29%) (from 1.33%)
G4CrossSectionDataStore::GetIsoCrossSection (2.18%)

libG4Geometry counter: from 6000 to 8200

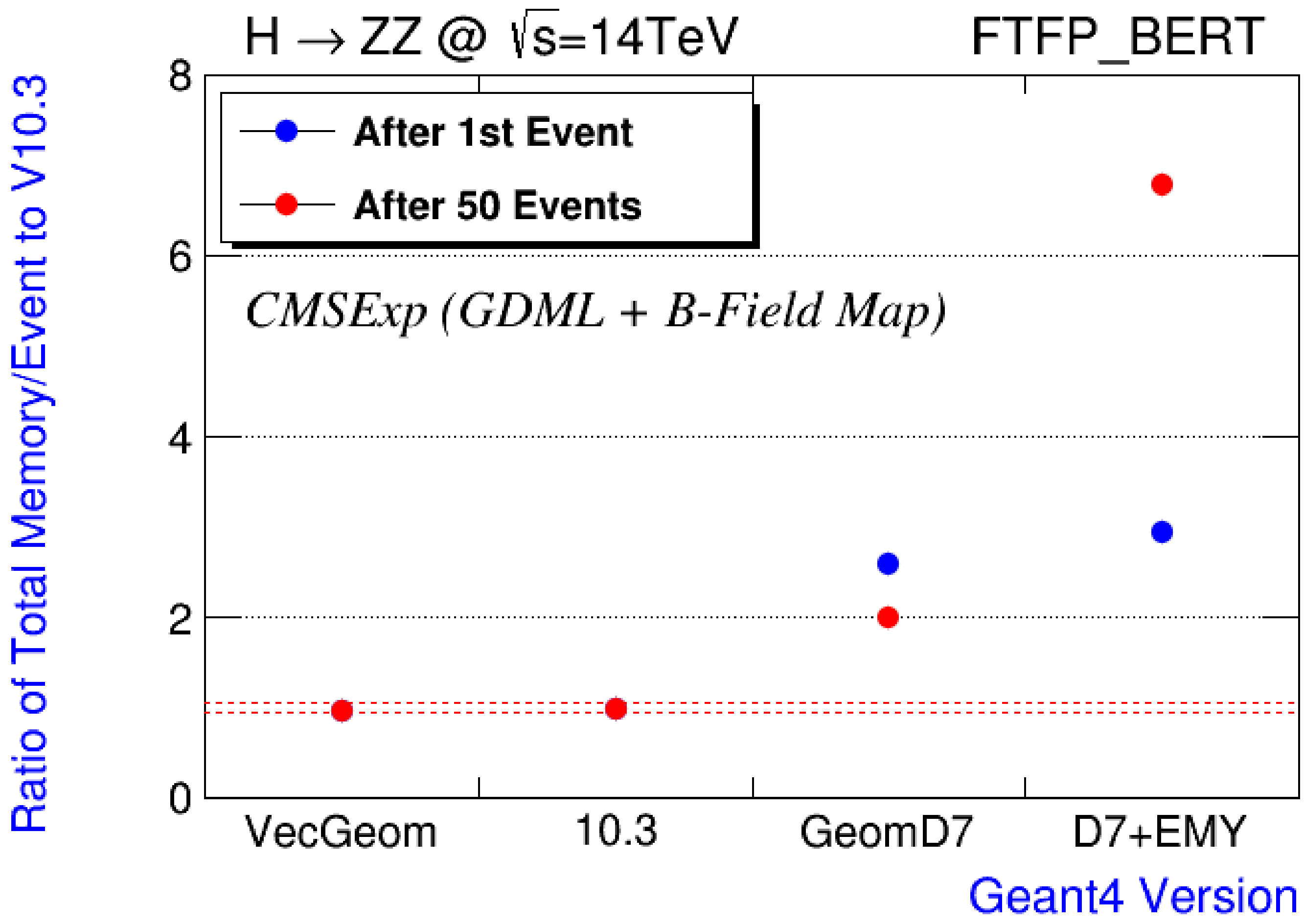
Additional 50% degradation from GeomD7 to EMY:

Top level functions counters are significantly increased
G4PhysicsVector::Value (from 143 to 246)
CLHEP::MTwistEngine::flat (from 52 to 98)
G4UniversalFluctuation::SampleFluctuations (from 37 to 93)

libProcesses counter: from 6100 to 11500

cmsExp Computing Performance with Geant4 10.3

- Total Memory Counter: Performance is normalized to 10.3



(detail profiling information at https://g4cpt.fnal.gov/)

Memory Performance (After 50 Events)

- There is almost no change in memory footprint with VecGeom
- GeomD7 (x2 increase in the total memory counter)
  - total memory counter from 9.9 GB to 19.7 GB:
    G4BoundingEnvelope::CreateListOfEdges (3.5 to 10.1 GB)
    G4BoundingEnvelope::CalculateExtent (0.6 to 1.5 GB)
  - VSIZE: from 390 to 720 MB
  - RSS : from 280 to 610 MB
- GeomD7+EMY (x3.4)
  - Total memory counter changed from 19.7 GB to 67.1 GB
    G4AtomicTransitionManager::Shell (41.3 GB)
    G4AtomicTransitionManager::ReachableShell (4.9 GB)
  - VSIZE: from 720 to 1250 MB
  - RSS : from 610 to 1150 MB

Summary

- Run-2 production of CMS is done using Geant4 10.0p02
  - 16 billion events are produced
- 2017 CMS full simulation switches to Geant4 10.2p02
  - Validated using test-beam and Run-2 data
- We are starting a set of benchmarks for evaluation of physics accuracy and CPU/memory profiling for HL-LHC
  - Standalone using only CMS gdml geometry and field
  - CMSSW CPU/Memory
  - CMSSW test-beam applications
  - CMSSW comparisons versus Run-2 data
- We need to establish benchmark conditions allowing comparisons across different platforms and experiments
  - Set of architectures
  - Set of Physics Lists
  - Primary generators
  - Statistics