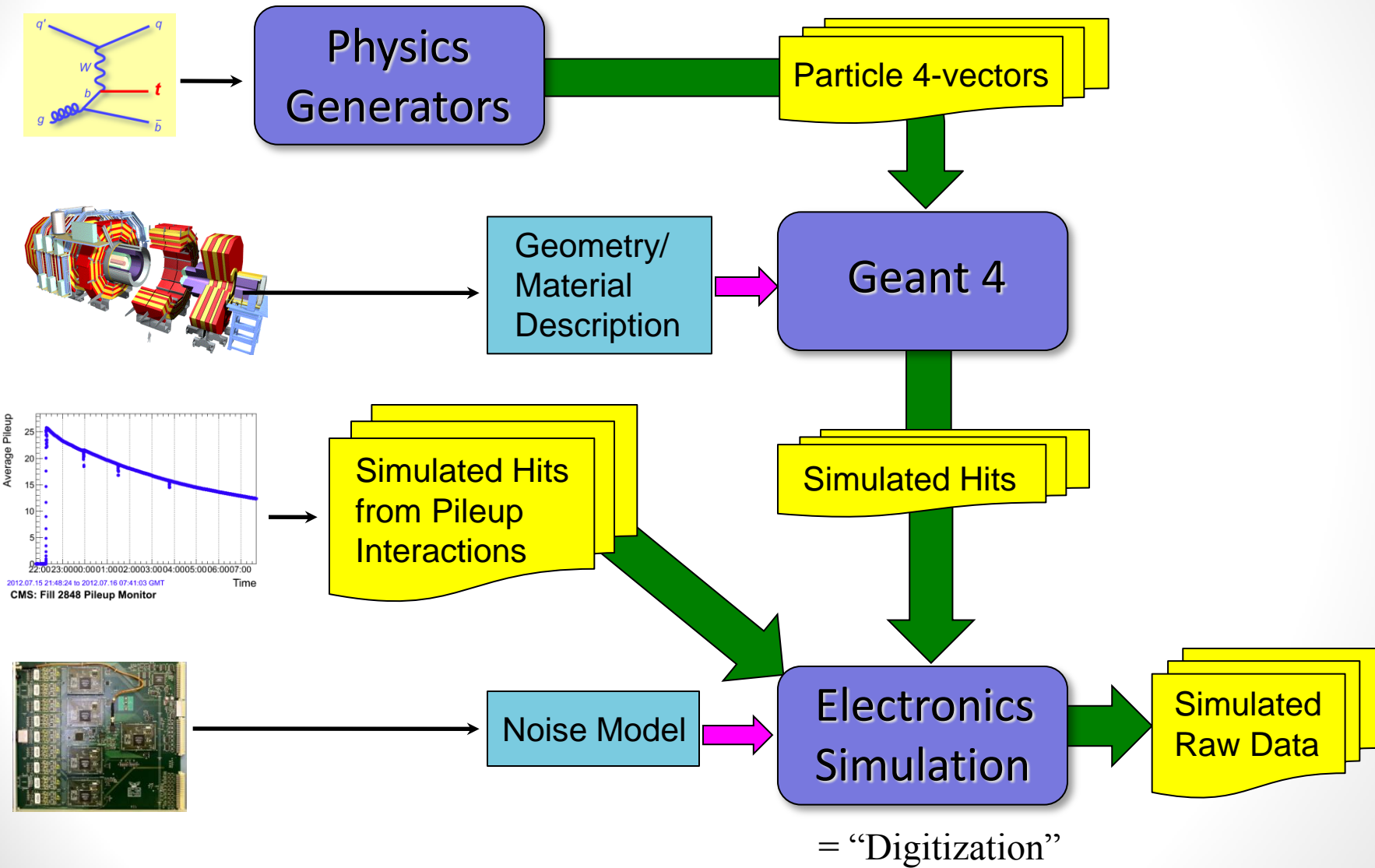


CMS Full Simulation for Run-2

M. Hildrith, V. Ivanchenko, D. Lange

CMS MC Simulation approach



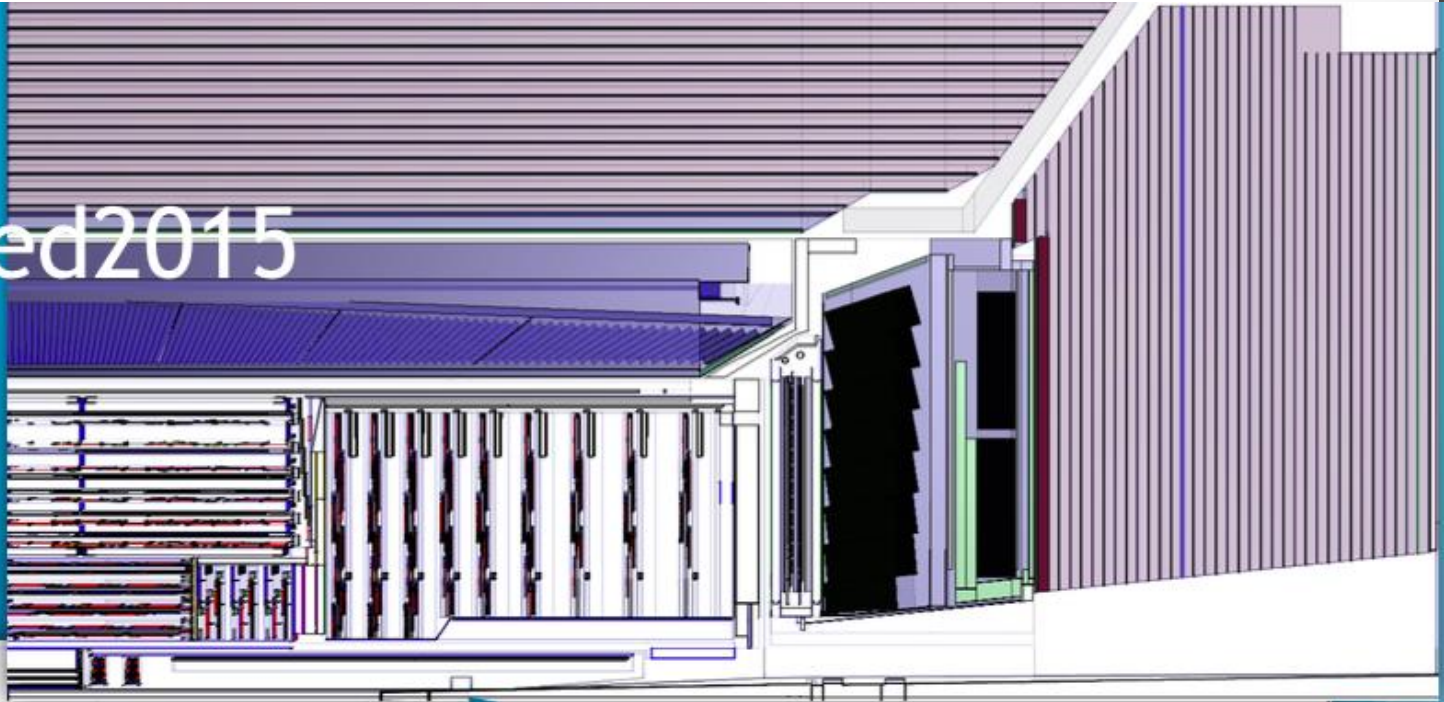
Progress in simulation for Run-2

- Geometry updates: Improvements to Run 1 detector model and integration of Run 2 detector changes
- Migration from Geant4 9.4p03 to 10.0p02
- Revised CMS simulation code
- Introduced Russian roulette method
- New forward hadron calorimeter parameterized simulation (shower library instead of GFlash)
- Improved simulation library packaging
- Improved mixing module for high pile-up simulation
- Developed simulation for multi-threaded CMSSW framework

Together these improvements have gained a factor of 2 in simulation time/event

CMS geometry for run-2

Extended 2015



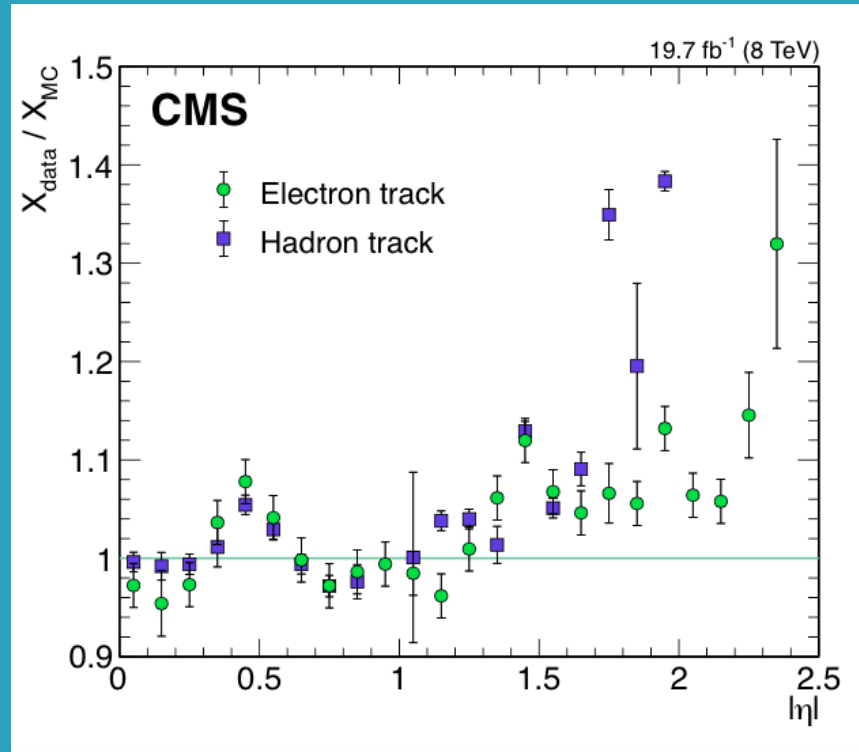
- New beam pipe model
- Updated muon detectors
- Improved geometry of tracker
- New set of lumi detectors in FWD

Update of tracker material model description

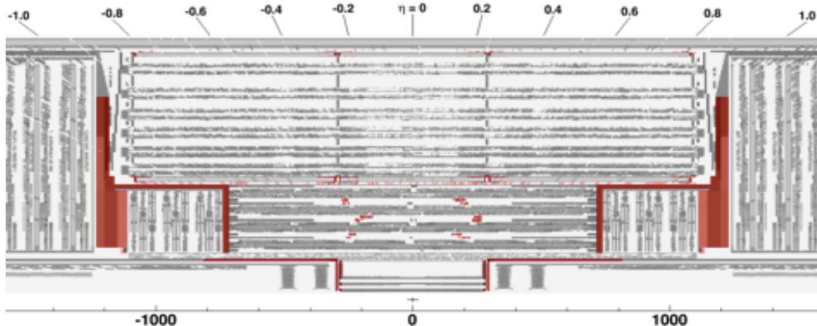
Tracker material model corrected to better match Run 1 data

- Measurement of hadron momentum loss in tracker volume
- Difference in electron momentum measured in forward vs backward track fit estimates Bremsstrahlung

Ratio of data vs. MC tracker material



Changes in our tracker material model



Geant4 status in CMS

- Production version of Geant4 for run-2
 - Geant4 version 10.0p02 built in sequential mode
 - Production platform slc6_amd64_gcc491
 - Default physics list: **QGSP_FTFP_BERT_EML**
 - CMS plans a production of about 4 billion events for 2015
- Current development version of Geant4 in CMS
 - Geant4 version 10.0p03 + patch of Geant4e for threading
 - Multi-threaded Geant4 is fully integrated with CMS multi-threaded framework

From Geant4 9.4p03 to 10.0p02

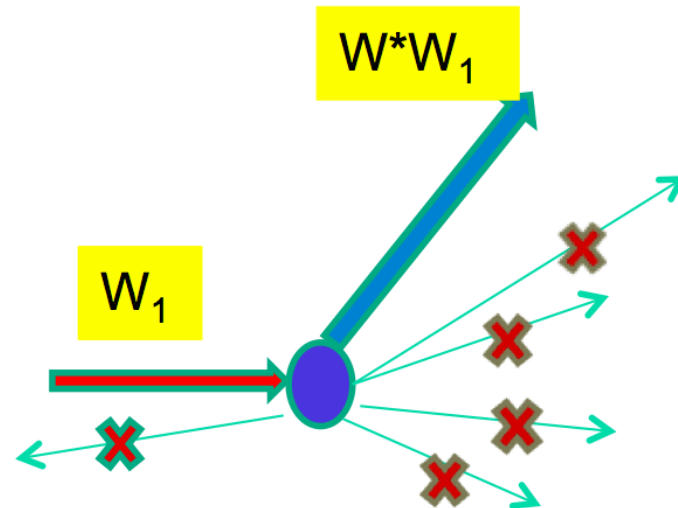
- More accurate electromagnetic shower shape simulation
- Improved hadronic physics simulation
- Performance improvements

Run 2 changes for Simulation

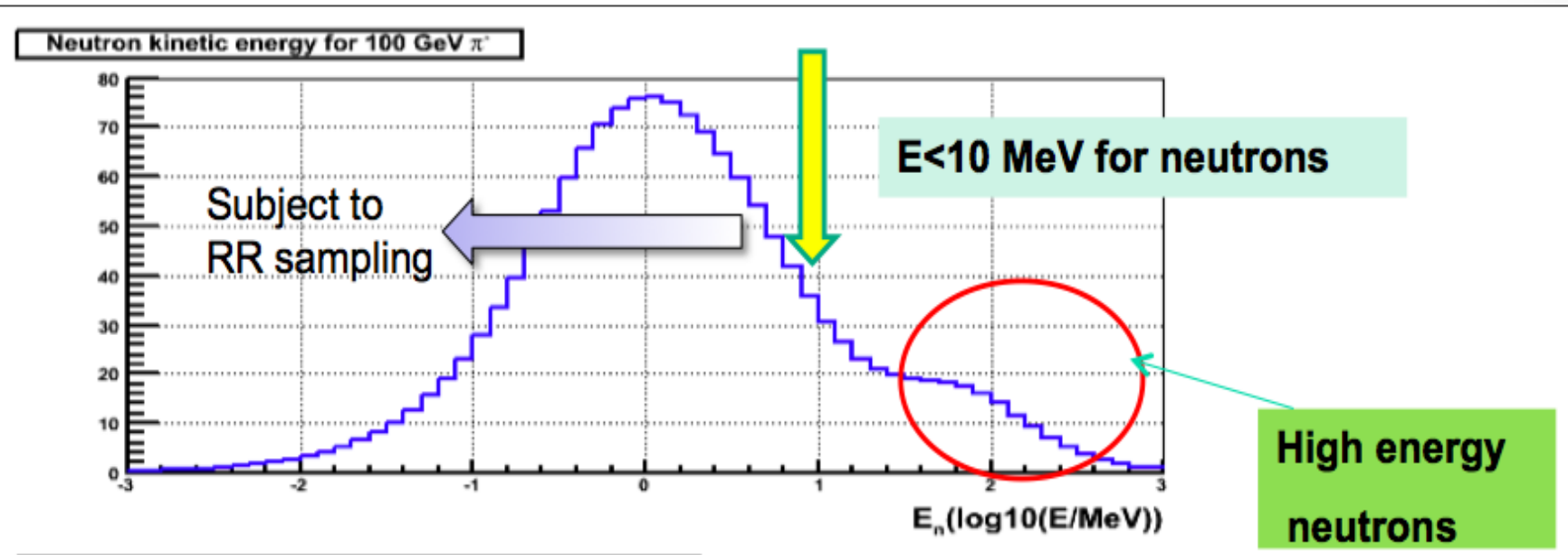
- Increasing of energy from 8 to 13 TeV provides more simulation steps and more hits in detectors
- User actions were optimized and rewritten
- Physics List QGSP_FTFP_BERT_EML provides similar detector response in hadronic calorimeter
- Validation of CMSSW demonstrates that EM shower shape was changed due to reduced number of secondary low-energy neutrons produced in gamma-nuclear interactions

Russian Roulette: Sampling of low-energy particles in Geant4

- Method from neutron shielding calculations: Track only a small fraction of low-energy particles through the detector with no noticeable change in simulation results
 - We found that it was necessary to have sampling factors and thresholds that depend on both detector region and particle type.
- Two parameters:
 - RR factor ($1/W$): Fraction of particles to keep
 - Upper energy limit (E_{RR})
- Hits from Particles below E_{RR} that are tracked are given a weight W .

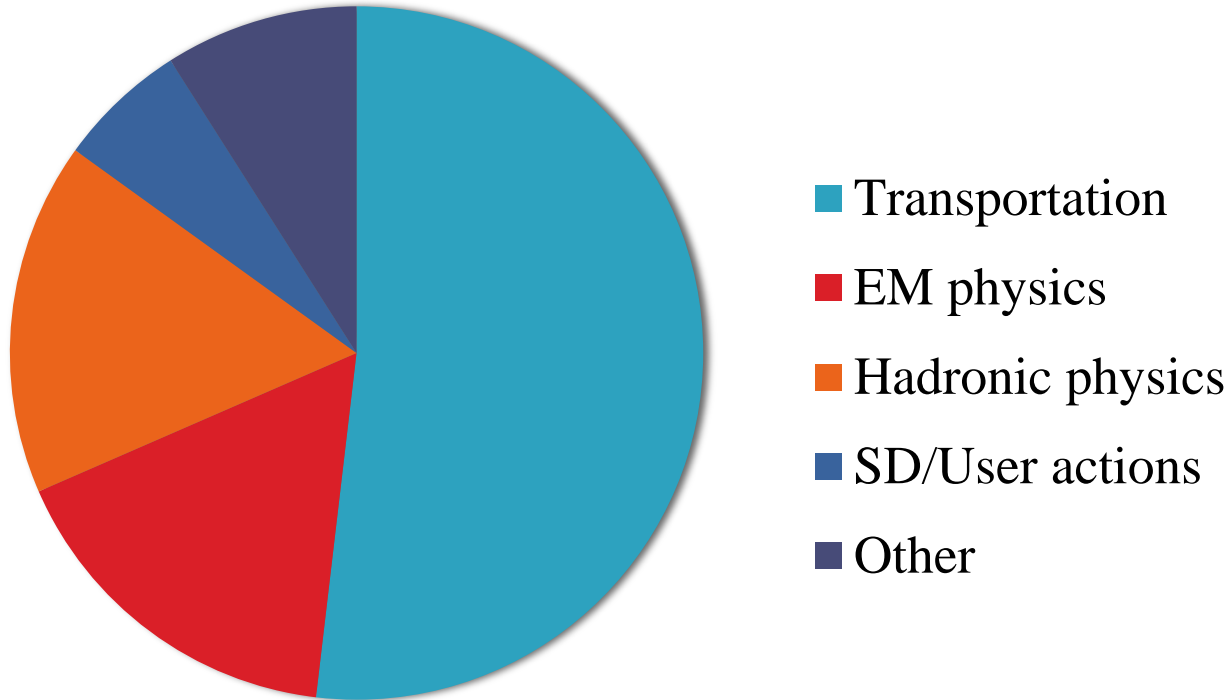


Russian Roulette now used by default after long tuning and validation process



- RR factor of $W=10$ for neutrons and 3 for gammas found to give between 25% and 40% performance improvement with no observable effect on physics output
 - Energy and shower shape response in the high-resolution ECAL barrel detector were the most sensitive to RR parameter tuning

Where our simulation CPU goes



- Technical performance improvements for Run 2 simulation:
 1. Upgrade to Geant4 10.0p03 (~5%)
 2. Implementation of Russian Roulette technique (~30%)
 3. CMSSW code optimization (~15%)
 4. Library repackaging (~10%)

10% performance gain from hidden visibility without playing with linker scripts

Repackage all shared libraries in CMSSW that depend on Geant4 into a single static library to “hide” Geant4 from the rest of CMSSW

- Use single archive library for Geant4 itself
- This allowed us to more aggressively optimize at link time: adding “**-flto -Wl,--exclude-libs,ALL**” works best

Constraints this imposes:

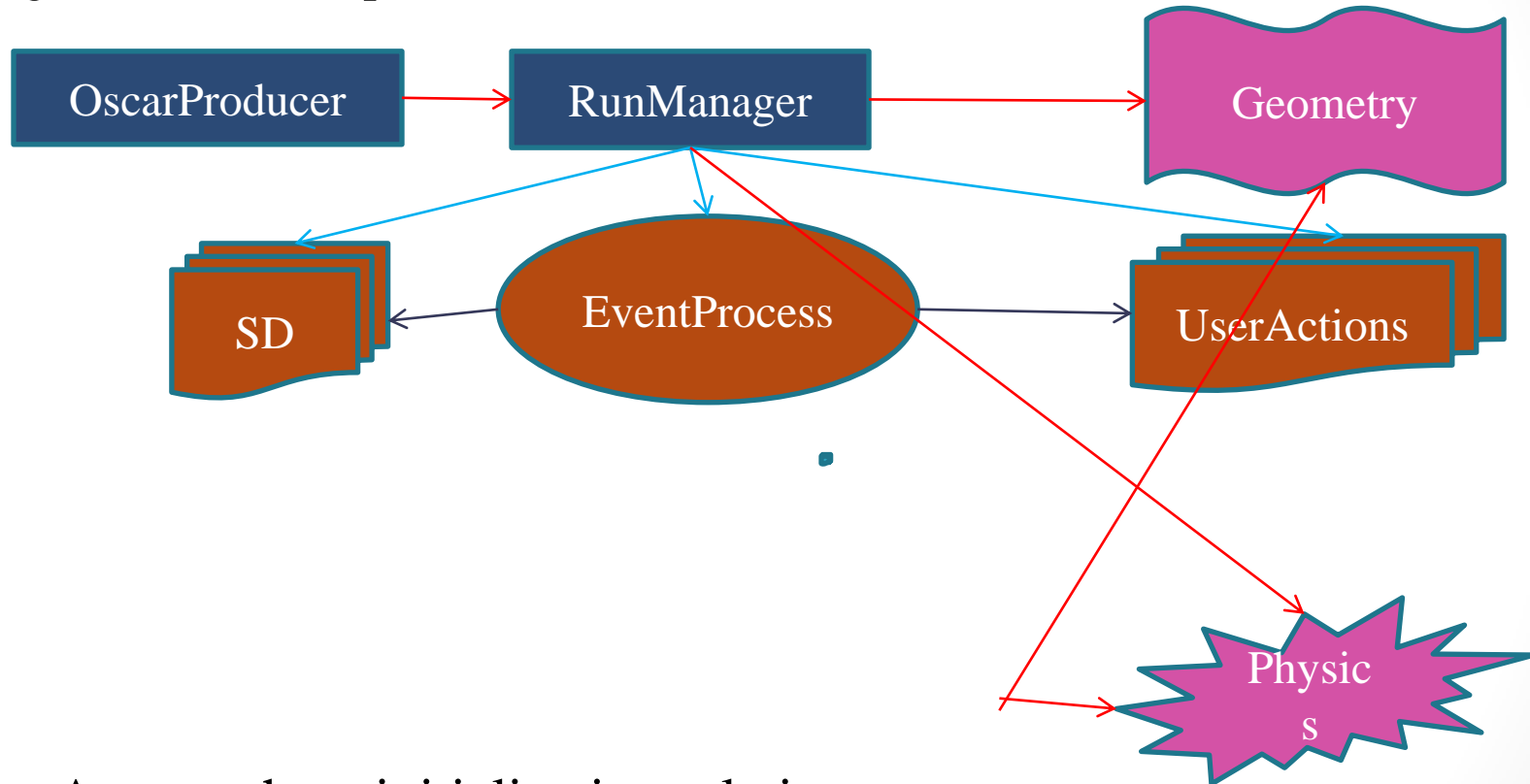
- Must control dependencies to use Geant4 only within this single library:
 - This is “easy” for simulation (<2% of our libraries)
 - However, extending this idea to something effecting the full reconstruction is difficult
- Impact on simulation code developers minimized by keeping .so cached in release. Static library rebuild is the only extra step if developer builds a package in this static library.

CMS multithreaded simulation

- **CMS has developed and deployed a multi-threaded framework based on TBB. Our simulation workflow is an important user**
 - At the same time, we need to work within the Geant-4 multithreading model (POSIX threads)
- **Master vs worker threads**
 - Master must not do any work
 - Worker threads simulate events in their own event loops (Spawned by CMSSW Framework via TBB)
- **Constraints and requirements we worked through:**
 - Initialization of SIM module should be done once
 - Access to geometry and field should be const
 - Both Sequential and MT simulation should be functional within a CMSSW release

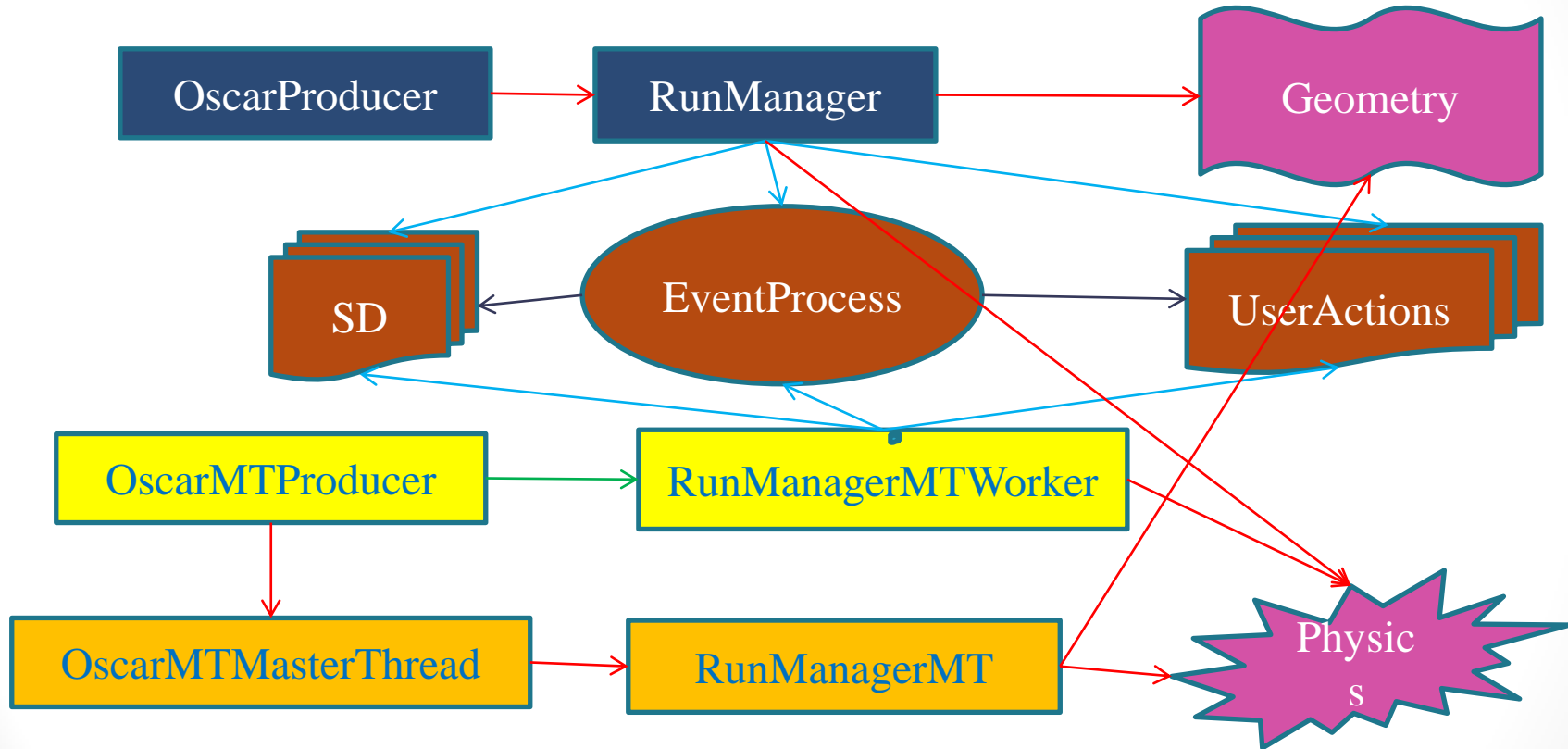
CMS simulation schemas

Single threaded (sequential) schema:



- Arrows show initialization relations

CMS simulation schemas



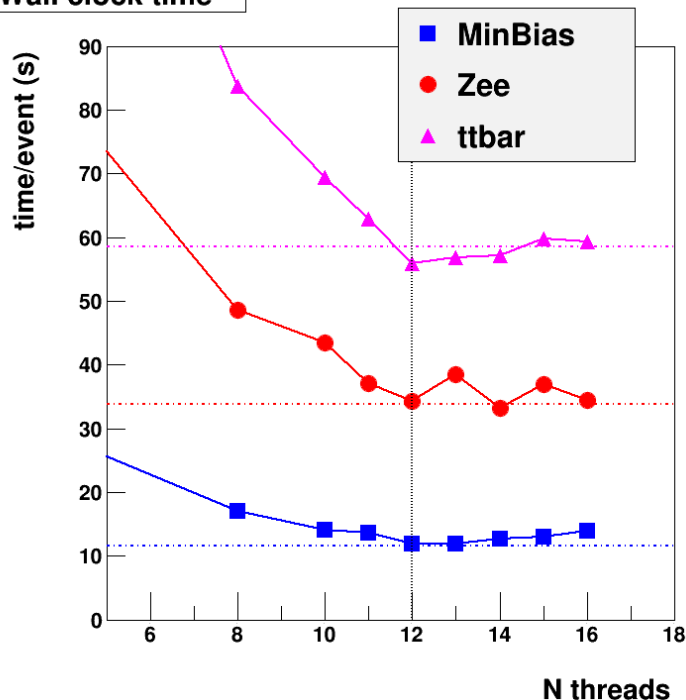
- Main difference between sequential and MT schemas is in initialization
- Geometry, physics, user actions, and SD share the same code

Current performance of CMS MT GEN-SIM



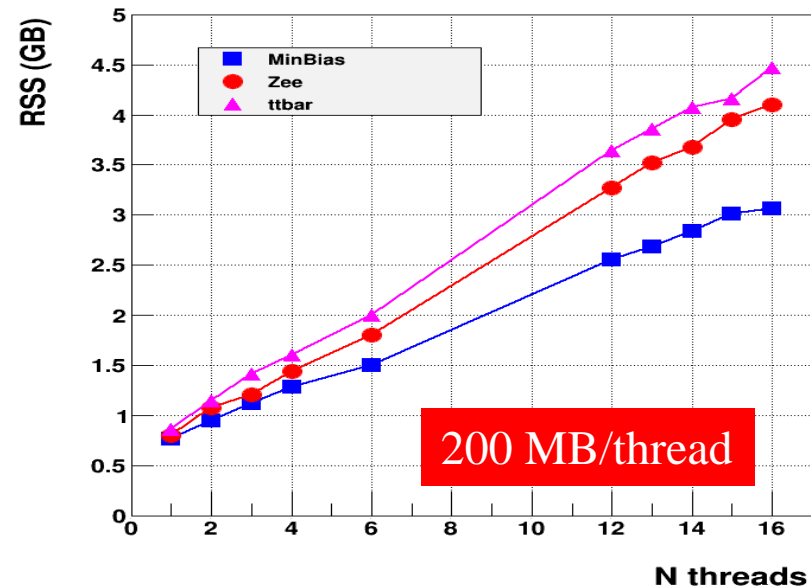
Time/event decreases until the # of threads is equal to the # of cores

Wall clock time



--- Extrapolation from single core

Example memory savings: a single 12 threaded MT job requires **~4 GB** RSS instead of **11 GB** for 12 single threaded jobs



Excellent scaling performance seen in our tests so far. Scale testing to start for production in 2016

Conclusion: Looking forward

- We are always looking at how to do more simulation with less resources without giving up physics performance
 - Large gains achieved since Run 1. Now investigating other areas including interesting new components of Geant4
 - Deployment of multithreaded simulation planned for 2016
- Supporting planned Phase-I and Phase-II CMS upgrades pose new challenges.