Fast detector simulations for LHC experiments

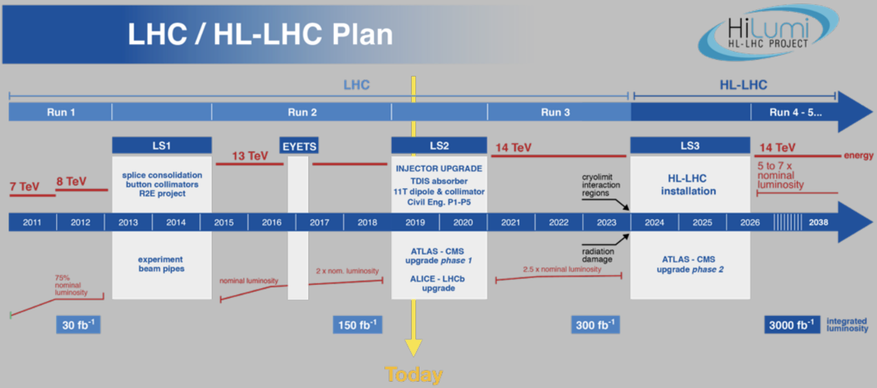
LHCP 2019, Puebla, Mexico

Sam Bein, on behalf of the LHC experiments

University of Hamburg

May 23, 2019

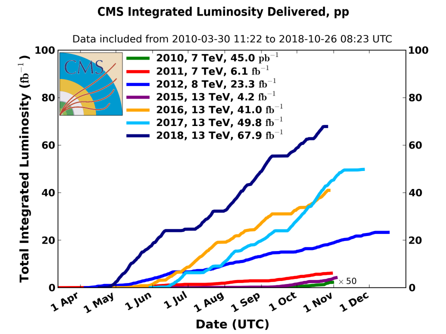
LHC Timeline

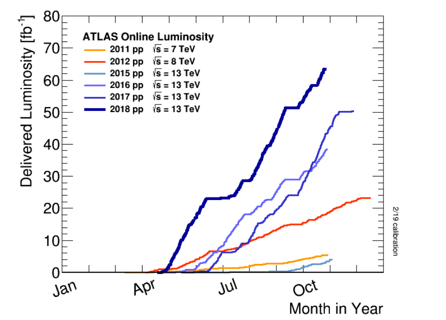


- We are 30% of the way through with the nominal LHC/HL-LHC timeline
- We have collected 5% of the total aspired dataset

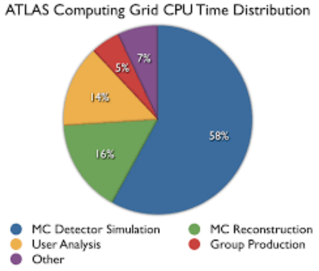
Luminosity by year

- Luminosity doubling roughly every 2 to 3 years
- Ever-increasing demand on resources, especially simulation



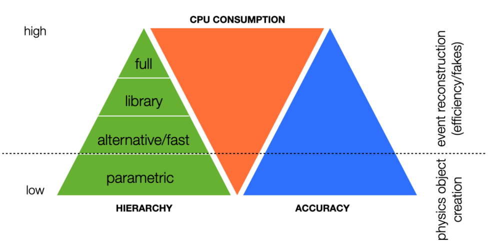


CPU usage



ATLAS Computing-link

Essential tradeoff



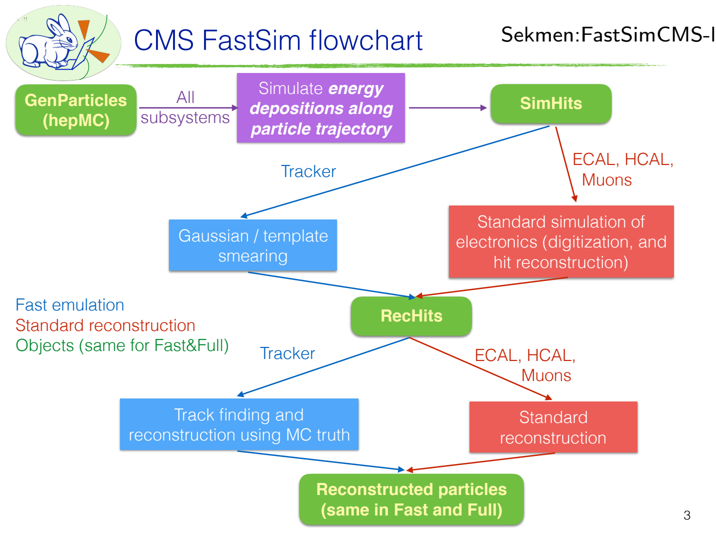
J. Phys.: Conf. Ser. 664 072024

Basic universal procedure

event generation → detector simulation → signal digitization → object reconstruction

CMS Fast Sim workflow

Sekmen:FastSimCMS-link



CMS FastSim flowchart

GenParticles (hepMC)

All subsystems

Simulate ***energy depositions along particle trajectory***

SimHits

Tracker

ECAL, HCAL, Muons

Gaussian / template smearing

Standard simulation of electronics (digitization, and hit reconstruction)

RecHits

Fast emulation
Standard reconstruction
Objects (same for Fast&Full)

Tracker

ECAL, HCAL, Muons

Track finding and reconstruction using MC truth

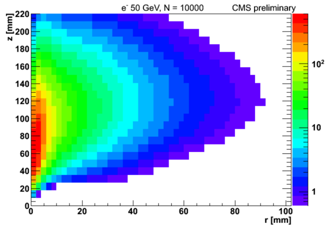
Standard reconstruction

Reconstructed particles (same in Fast and Full)

Fast Sim key components

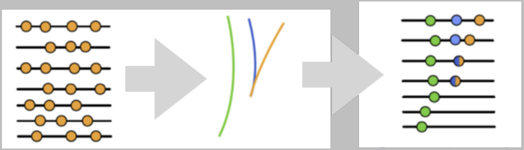
- simulation step - 100x speedup from GEANT4 based full simulation
  - simplified, modular detector geometry
  - energy-dependent parameterization of shower properties
  - parameters extracted from template fits to full simulation
  - tracker sim hits → Gaussian smearing → rec hits

shower parametrization



Pedro:FastCaloCms-link

fast simplified tracking



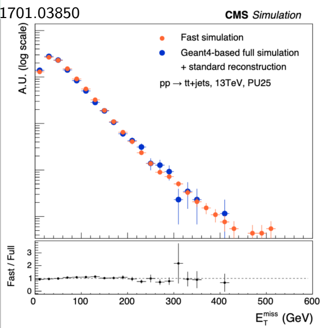
arXiv:1701.03850

- Reconstruction (FastSim tracking)
  - sim hits created at intersection points, smeared
  - sim hit mapping to generator particles saved
  - standard seeding algorithm performed on mapped subset
  - no fakes in the tracking, no hit merging or hit sharing among tracks

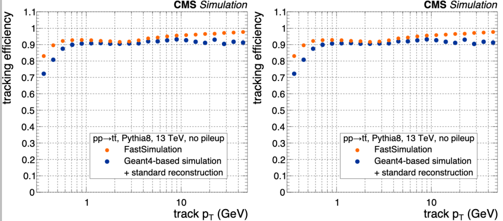
Fast Sim key components

- simulation step - 100x speedup from GEANT4 based full simulation
  - simplified, modular detector geometry
  - energy-dependent parameterization of shower properties
  - parameters extracted from template fits to full simulation
  - tracker sim hits → Gaussian smearing → rec hits

arXiv:1701.03850



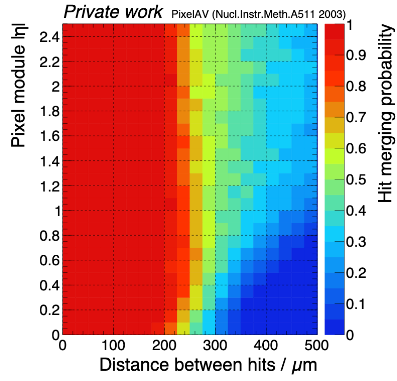
fast simplified tracking

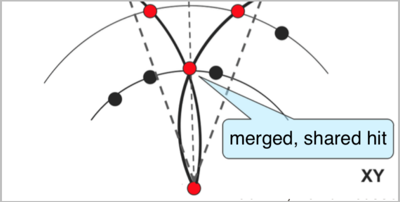


arXiv:1701.03850

- Reconstruction (FastSim tracking)
  - sim hits created at intersection points, smeared
  - sim hit mapping to generator particles saved
  - standard seeding algorithm performed on mapped subset
  - no fakes in the tracking, no hit merging or hit sharing among tracks

CMS FastSim: around the corner





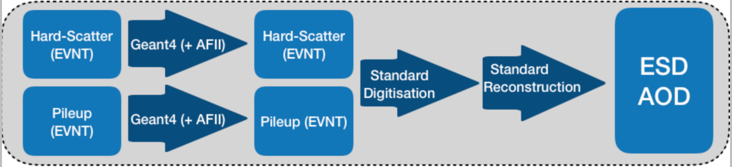
- merged hit probability parameterized vs proximity and η
- maps applied during SIM step
- dropped 1-1 mapping between hits and true particles → final tracks share a single merged hit

J. Phys.: Conf. Ser. 898 (2017)

ATLAS Fast Sim workflow

Full simulation chain

Pascuzzi:FastSim-link



Hard-Scatter (EVNT)

Geant4 (+ AFII)

Hard-Scatter (EVNT)

Pileup (EVNT)

Geant4 (+ AFII)

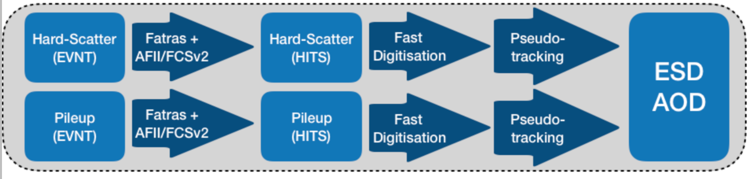
Pileup (EVNT)

Standard Digitisation

Standard Reconstruction

ESD AOD

Fast simulation chain



Hard-Scatter (EVNT)

Fatras + AFII/FCSv2

Hard-Scatter (HITS)

Fast Digitisation

Pseudo-tracking

Pileup (EVNT)

Fatras + AFII/FCSv2

Pileup (HITS)

Fast Digitisation

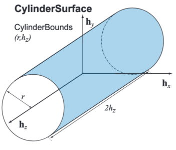
Pseudo-tracking

ESD AOD

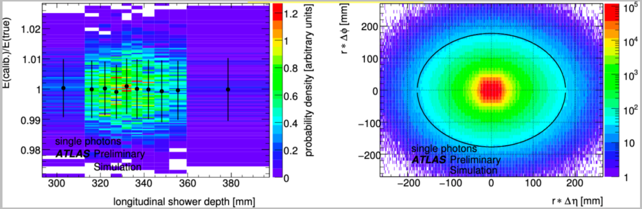
ATLAS fast sim key components

- Fast calorimetry - ATLFAST-II
  - simplified geometry, calo cells as $\eta - \phi$ cuboids
  - particle shower development parameterized (lateral and longitudinal)
  - O(10) reduction in CPU

ATL-PHYS-PUB-2010-013-link



- Geant4-based parameterizations
  - longitudinal profile
  - lateral profile

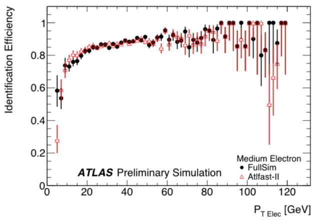


→Current using full sim tracking, digitization

ATL-SOFT-PUB-2014-001-link

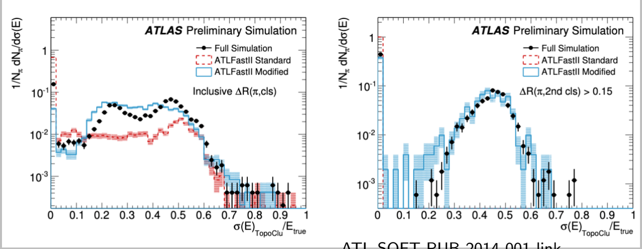
ATLAS fast sim key components

ATL-PHYS-PUB-2010-013-link



- Fast calorimetry - ATLFAST-II
  - simplified geometry, calo cells as $\eta - \phi$ cuboids
  - particle shower development parameterized (lateral and longitudinal)
  - O(10) reduction in CPU

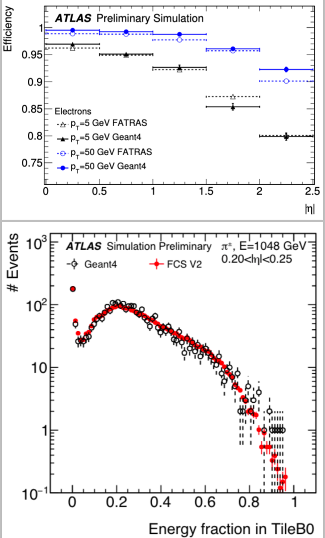
- Geant4-based parameterizations
  - longitudinal profile
  - lateral profile



ATL-SOFT-PUB-2014-001-link

→Current using full sim tracking, digitization

ATLAS ATLFAST-IIF: around the corner

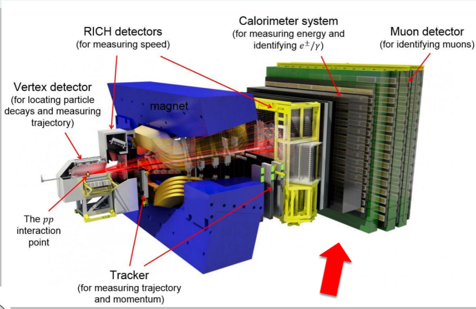


ATL-SOFT-PUB-2018-002-link

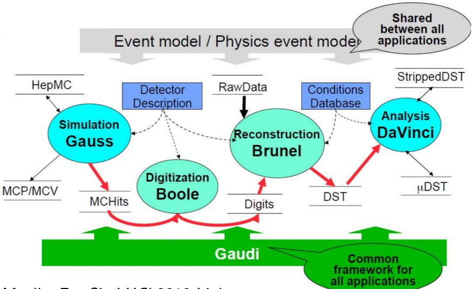
- Fast tracking - FATRAS, part of ATLFAST-IIF
  - produces hits and tracks
  - material effects based on traversed material
  - multiple scattering, bremmsstrahlung, photon
  - O(10) reduction in CPU
- Fast calorimetry - FastCaloSimV2
  - regression-based longitudinal shower modeling
  - improved lateral shower modeling
  - energy and shower shape parameterisations
  - O(10) reduction in CPU

LHCb Simulation

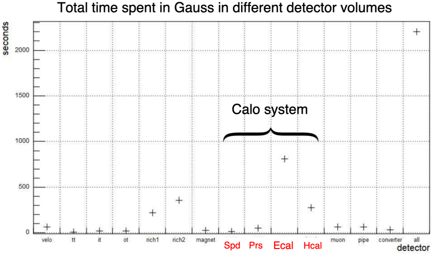
- all Run 1 and 2 analyses make use of GEANT4-based simulation
- techniques to speed the simulation up have been implemented
  - Redecay, Richless simulation, ...
- fast methods to sidestep G4 in development



RICH detectors
(for measuring speed)
Calorimeter system
(for measuring energy and identifying $e^{\pm}/\gamma$)
Muon detector
(for identifying muons)
Vertex detector
(for locating particle decays and measuring trajectory)
magnet
The pp interaction point
Tracker
(for measuring trajectory and momentum)

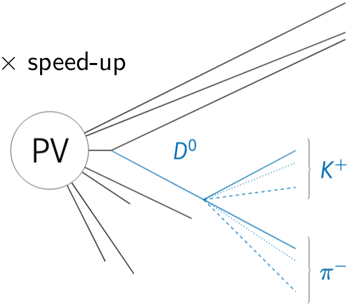


Mueller:FastSimLHCb2019-Link

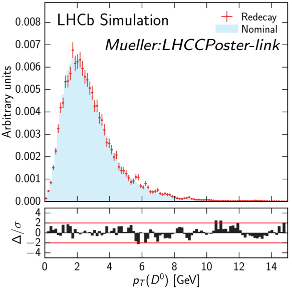


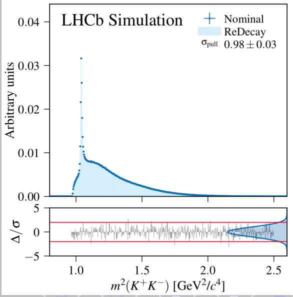
LHCb ReDecay (Geant4-based)

10-50 × speed-up



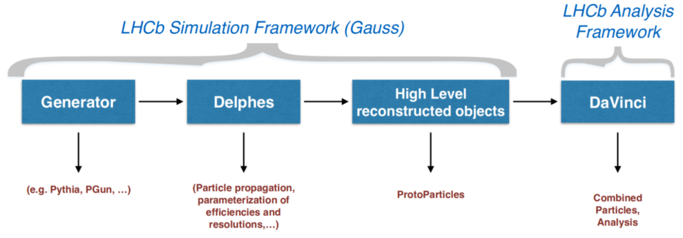
1. Generate full event.
2. Take out the signal.
3. Simulate the remaining event.
4. Generate and simulate multiple signals.
5. Combine simulated signal with stored underlying event.





LHCb fast sim around the corner

- Large speed-ups possible through parametrization



LHCb Simulation Framework (Gauss)

LHCb Analysis Framework

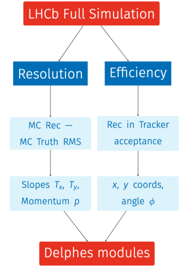
Generator → Delphes → High Level reconstructed objects → DaVinci

(e.g. Pythia, PGun, ...)

(Particle propagation, parameterization of efficiencies and resolutions,...)

ProtoParticles

Combined Particles, Analysis



LHCb Full Simulation

Resolution

Efficiency

MC Rec − MC Truth RMS

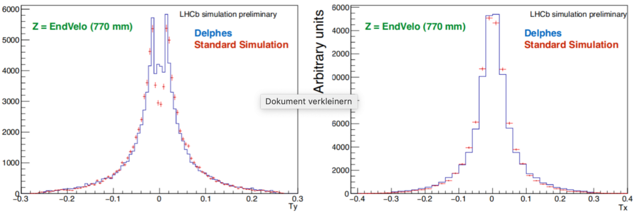
Rec in Tracker acceptance

Slopes T_x, T_y, Momentum p

x, y coords, angle ϕ

Delphes modules

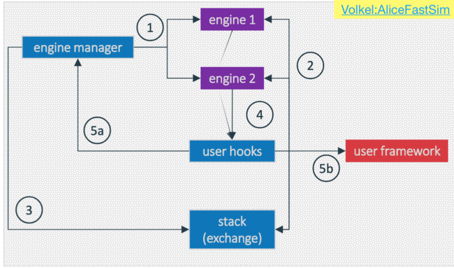
- Large speed-ups possible through parametrization



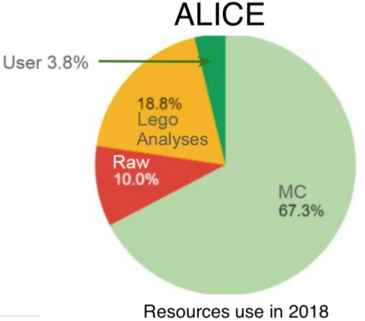
Mueller:LHCCPoster-link

ALICE simulation

- detector simulation based on ROOT VMC
- all analysis currently uses Geant4 based simulation
- plan is to mix multiple full simulation engines with fast simulation kernels
- fully parametrised fast simulation approach for upgrade studies



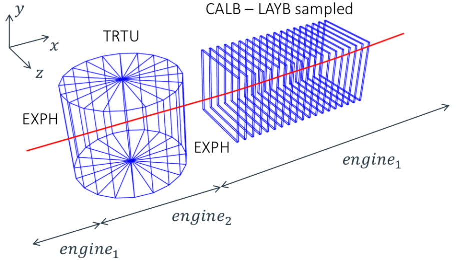
Volkel:AliceFastSim



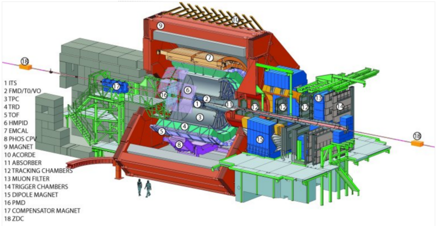
Resources use in 2018

Volkel:FastSimALICE-link

ALICE simulation proof of concept



19 detector sub-systems

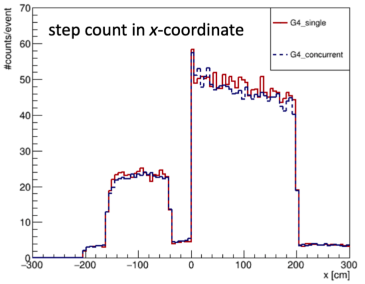


idea:

- fully simulate signal objects or specific sub-detector
- use fast simulation for the rest of the event
- test mixing of two simulation engines with toy model
- cross check with single and concurrent Geant4 instances

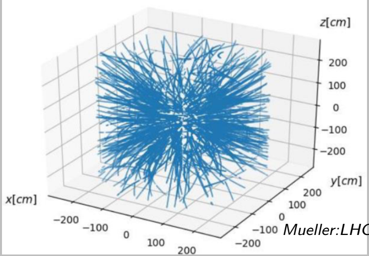
Volkel:FastSimALICE-link

Pascuzzi:FastSim-link2



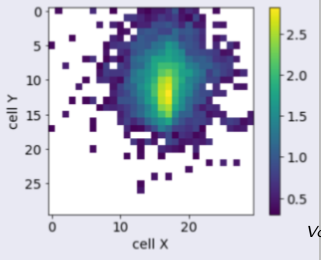
Machine learning and fast simulation

ALICE condDCGAN-generated event



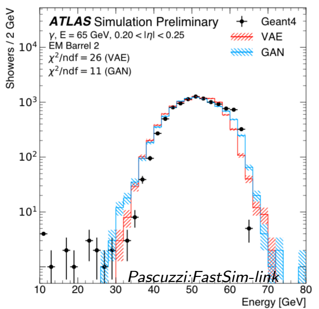
Mueller:LHCCPoster-link

LHCb GAN-based showering model



Volkel:FastSimALICE-link

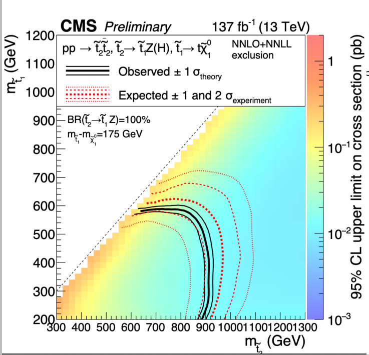
ATLAS VAE/GAN showing model

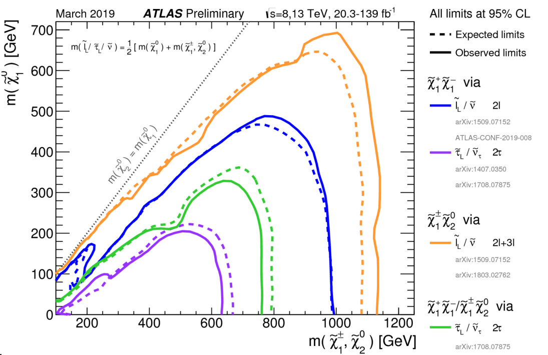


Pascuzzi:FastSim-link

Summary

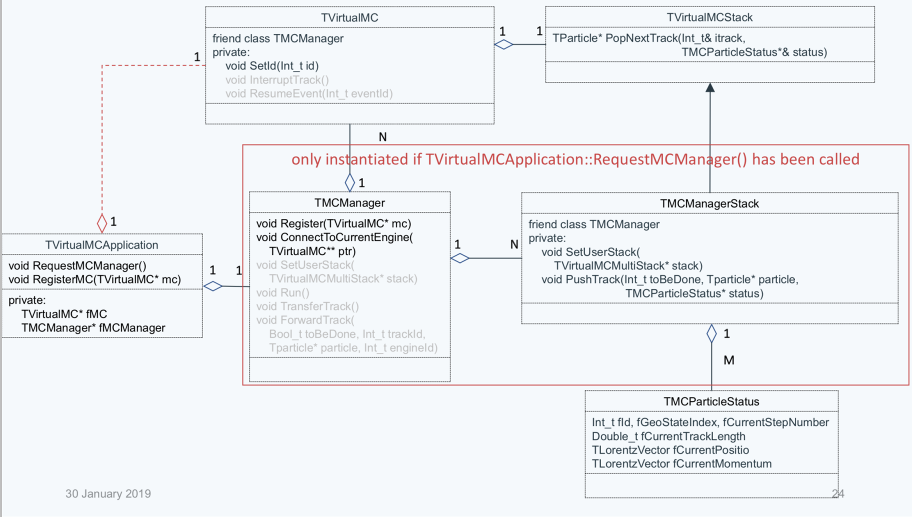
- Fast techniques have served the LHC experiments well in Runs 1 and 2
- Many developments are taking place in FastSim in both speed and performance
- These improvements are needed for keeping up with data and covering wider ranges of signal models
- Parameterization, libraries, fast tracking, machine learning are all candidates for filling this need





Backup

ALICE Faster SIM (i)



TVirtualMC

friend class TMCManager
private:
    void SetId(Int_t id)
    void InterruptTrack()
    void ResumeEvent(Int_t eventId)

TVirtualMCStack

TParticle* PopNextTrack(Int_t& itrack,
                        TMCParticleStatus*& status)

only instantiated if TVirtualMCApplication::RequestMCManager() has been called

TMCManager

void Register(TVirtualMC* mc)
void ConnectToCurrentEngine(
    TVirtualMC** ptr)
void SetUserStack(
    TVirtualMCMultiStack* stack)
void Run()
void TransferTrack()
void ForwardTrack(
    Bool_t toBeDone, Int_t trackId,
    Tparticle* particle, Int_t engineId)

TMCManagerStack

friend class TMCManager
private:
    void SetUserStack(
        TVirtualMCMultiStack* stack)
    void PushTrack(Int_t toBeDone, Tparticle* particle,
                   TMCParticleStatus* status)

TVirtualMCApplication

void RequestMCManager()
void RegisterMC(TVirtualMC* mc)

private:
    TVirtualMC* fMC
    TMCManager* fMCManager

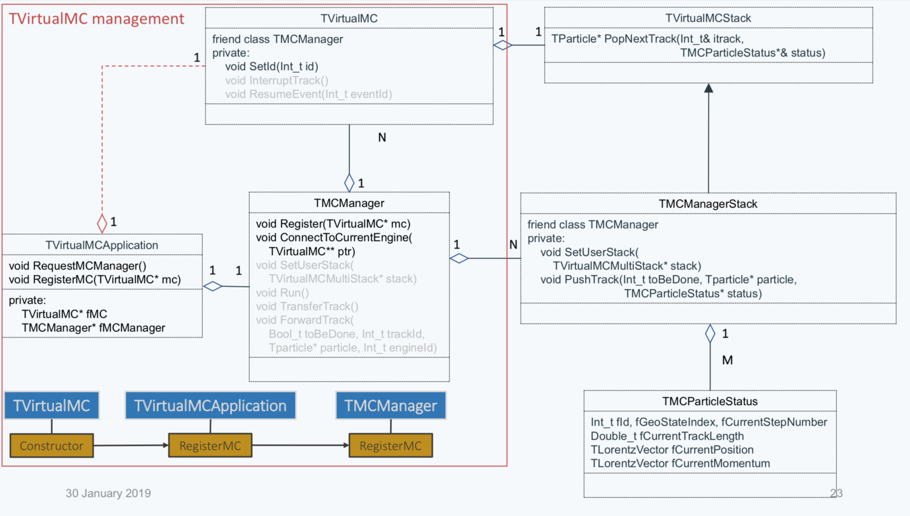
TMCParticleStatus

Int_t fId, fGeoStateIndex, fCurrentStepNumber
Double_t fCurrentTrackLength
TLorentzVector fCurrentPositio
TLorentzVector fCurrentMomentum

30 January 2019

24

ALICE Faster SIM (ii)



TVirtualMC management

```
TVirtualMC
friend class TMCManager
private:
   void SetId(Int_t id)
   void InterruptTrack()
   void ResumeEvent(Int_t eventId)
```

```
TVirtualMCStack
TParticle* PopNextTrack(Int_t& itrack,
                        TMCParticleStatus*& status)
```

```
TVirtualMCApplication
void RequestMCManager()
void RegisterMC(TVirtualMC* mc)
private:
   TVirtualMC* fMC
   TMCManager* fMCManager
```

```
TMCManager
void Register(TVirtualMC* mc)
void ConnectToCurrentEngine(
   TVirtualMC** ptr)
void SetUserStack(
   TVirtualMCMultiStack* stack)
void Run()
void TransferTrack()
void ForwardTrack(
   Bool_t toBeDone, Int_t trackId,
   Tparticle* particle, Int_t engineId)
```

```
TMCManagerStack
friend class TMCManager
private:
   void SetUserStack(
      TVirtualMCMultiStack* stack)
   void PushTrack(Int_t toBeDone, Tparticle* particle,
                  TMCParticleStatus* status)
```

```
TMCParticleStatus
Int_t fId, fGeoStateIndex, fCurrentStepNumber
Double_t fCurrentTrackLength
TLorentzVector fCurrentPosition
TLorentzVector fCurrentMomentum
```

TVirtualMC → Constructor

TVirtualMCApplication → RegisterMC

TMCManager → RegisterMC

30 January 2019

23