Status of CEPC Software

Xingtao Huang

On behalf of the CEPC Software Working Group

The 2020 International Workshop on the High Energy Circular Electron Positron Collider

Oct. 26-28, 2020 Shanghai

Outline

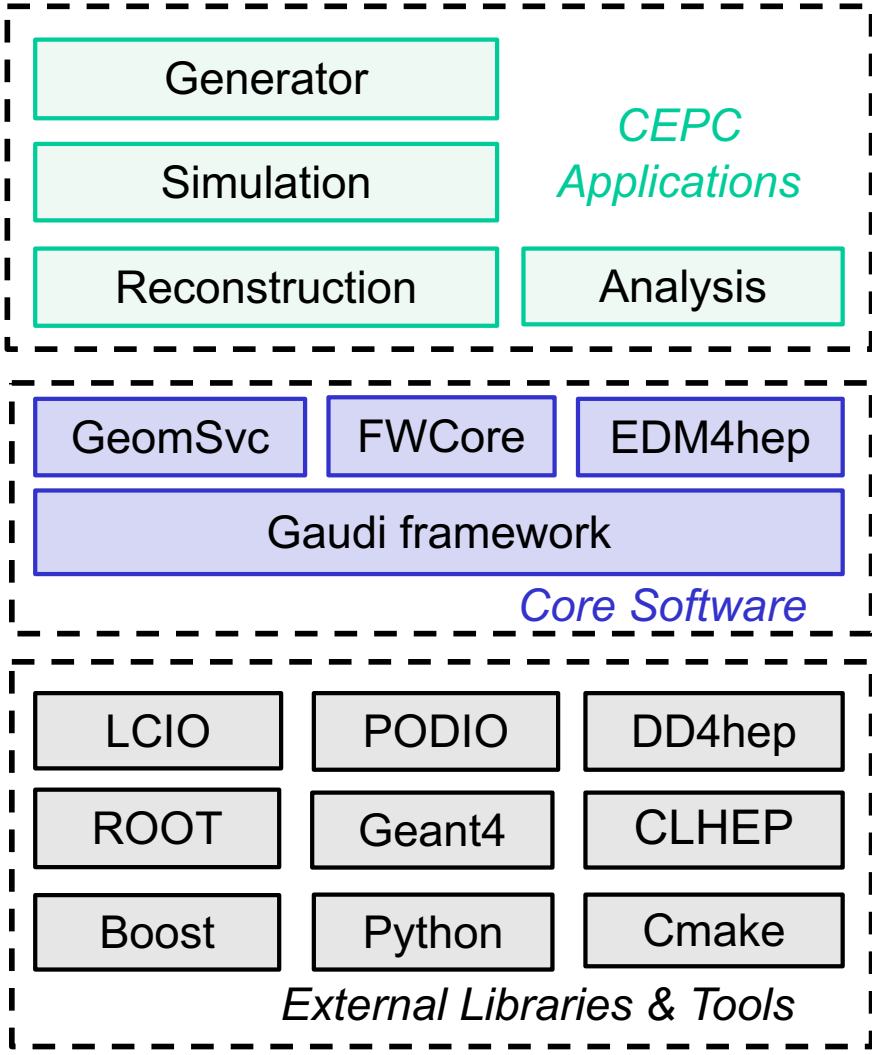
- Introduction
- CEPCSW and Core Software
- Event Data Model with EDM4hep
- Detector Description and Geometry management
- Simulation and Reconstruction (Tracker and ECAL)
- Reference Detector Study (Drift Chamber and ECAL)
- Software Environment and Development Workflow
- Summary & Plan

Introduction

- CEPC software originally started from the iLCSoft (many thanks)
  - Marlin, LCIO, MokkaC, Gear
  - Developed the components for CEPC: simulation, reconstruction…
  - Massive M.C. Data production for detector and physics study
  - Used for the CDR released in Nov, 2018
- April 2019: proposed a new CEPC software (CEPCSW) prototype for TDR at the Oxford workshop
- June 2019: reached the consensus among CEPC, FCC, ILC, CLIC,… at the Bologna workshop
  - Develop a Common Turnkey Software Stack (Key4hep) for future collider experiments
  - Maximize the sharing of software components between experiments
- Now CEPCSW is fully integrated with Key4hep, and well supports application development

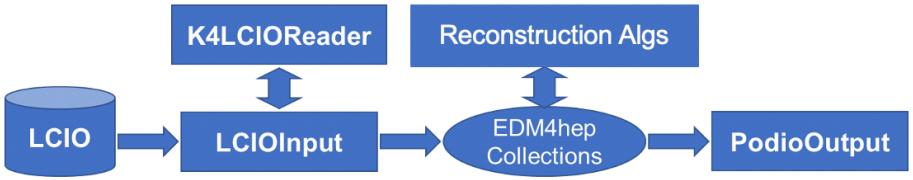
CEPCSW and Core Software

- Architecture of CEPCSW
  - External libraries
  - Core software
  - CEPC applications for simulation, reconstruction and analysis
- Core software
  - Gaudi framework: defines interfaces to all software components and controls their execution.
  - EDM4hep: generic event data model for HEP collider experiments
  - FWCore :manage event objects defined by EDM4hep.
  - GeomSvc :a DD4hep-based geometry service to provide a unified way to access detector geometry data.
  - Both FWCore and EDM4hep are Key4hep packages.

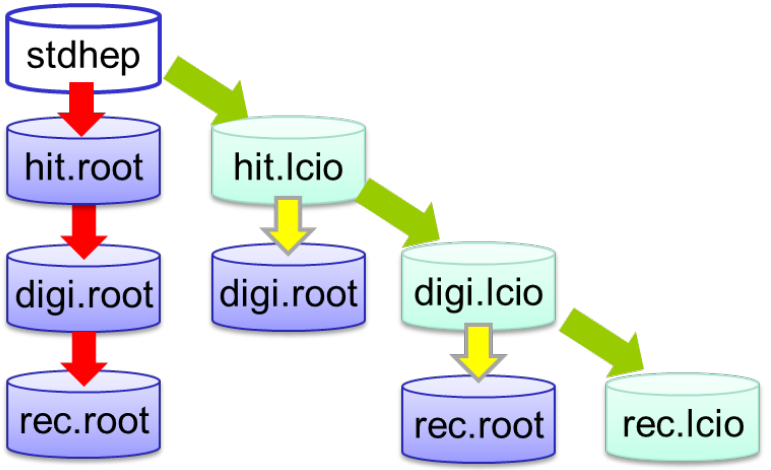


Event Data Model

- Originally using LCIO, then PLCIO
- Now fully adopted EDM4hep as the official EDM
- Developed new Key4hep components
  - K4LCIOReader
    - Converters: LCEvent → EDM4hep
  - LCIOInput
    - A Gaudi algorithm to register the converted data to the EDS
  - Very useful when porting algorithms from Marlin to CEPCSW and using the LCIO data to compare the results

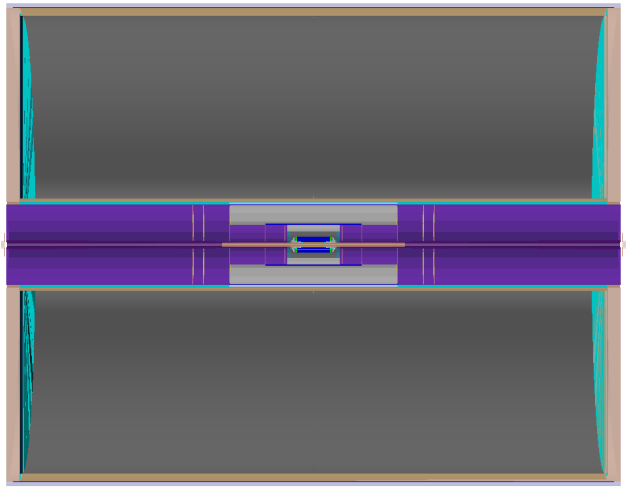


- **The Green arrows:**
  - LCIO+Mokka+Marlin
- **The Yellow arrows:**
  - Input: LCIO
  - Output: EDM4hep
- **The Red arrows:**
  - EDM4hep+DD4hep

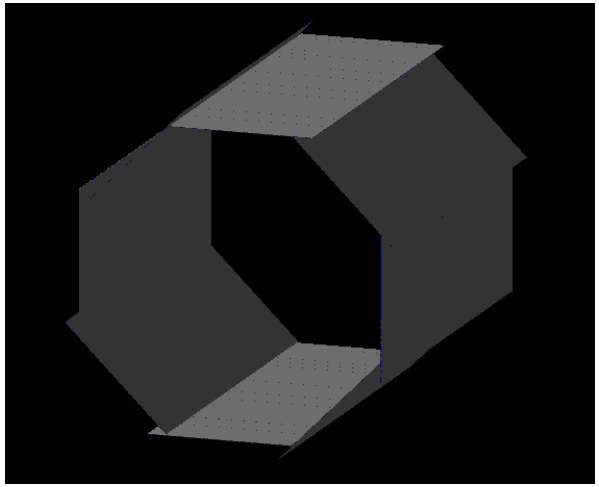


Detector Description in CEPCSW

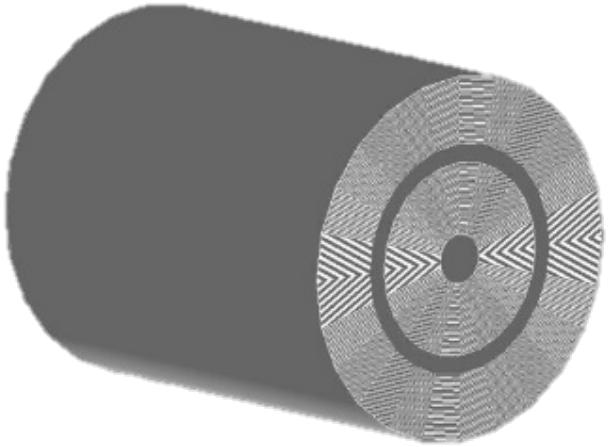
- The DD4hep toolkit is used to describe the detectors
  - The detector option CEPC_v4:
    - Trackers including VXD, FTD, SIT, TPC, SET
    - ECAL (Silicon +Tungsten)
  - The CEPC Reference Detector
    - Drift Chamber
    - ECAL (Crystal Bar)



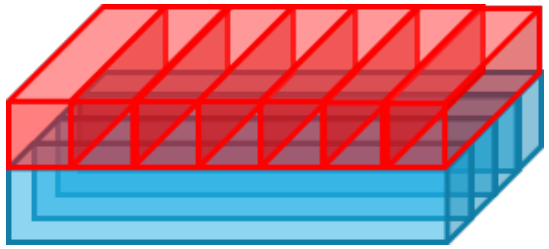
Trackers in CEPC_v4
- VXD
- FTD
- SIT
- TPC
- SET



One layer (Si+W+Si) of ECAL
- 8 staves
- 5 modules per stave
- 5 towers per modules



Drift Chamber
- Baseline:1.8m, 130 layers

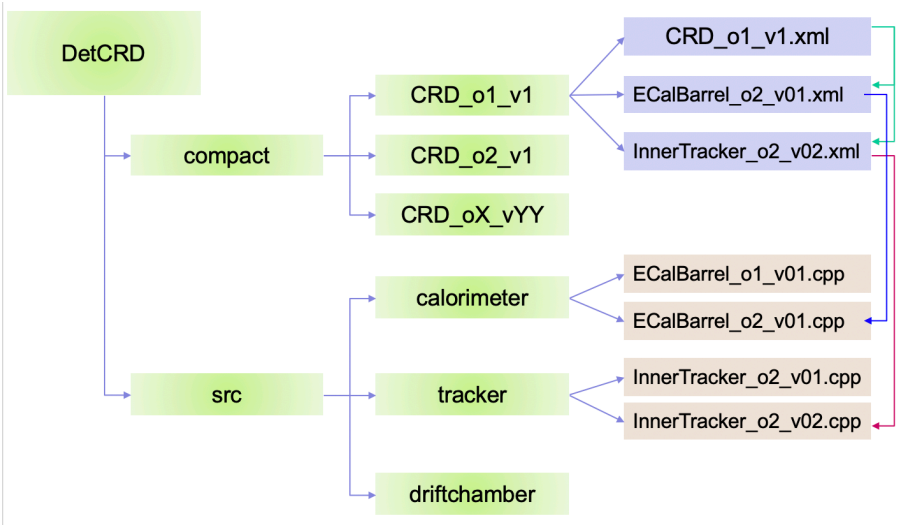


Super Cell

$40cm \times 40cm \times 2cm$

Detector Geometry Management

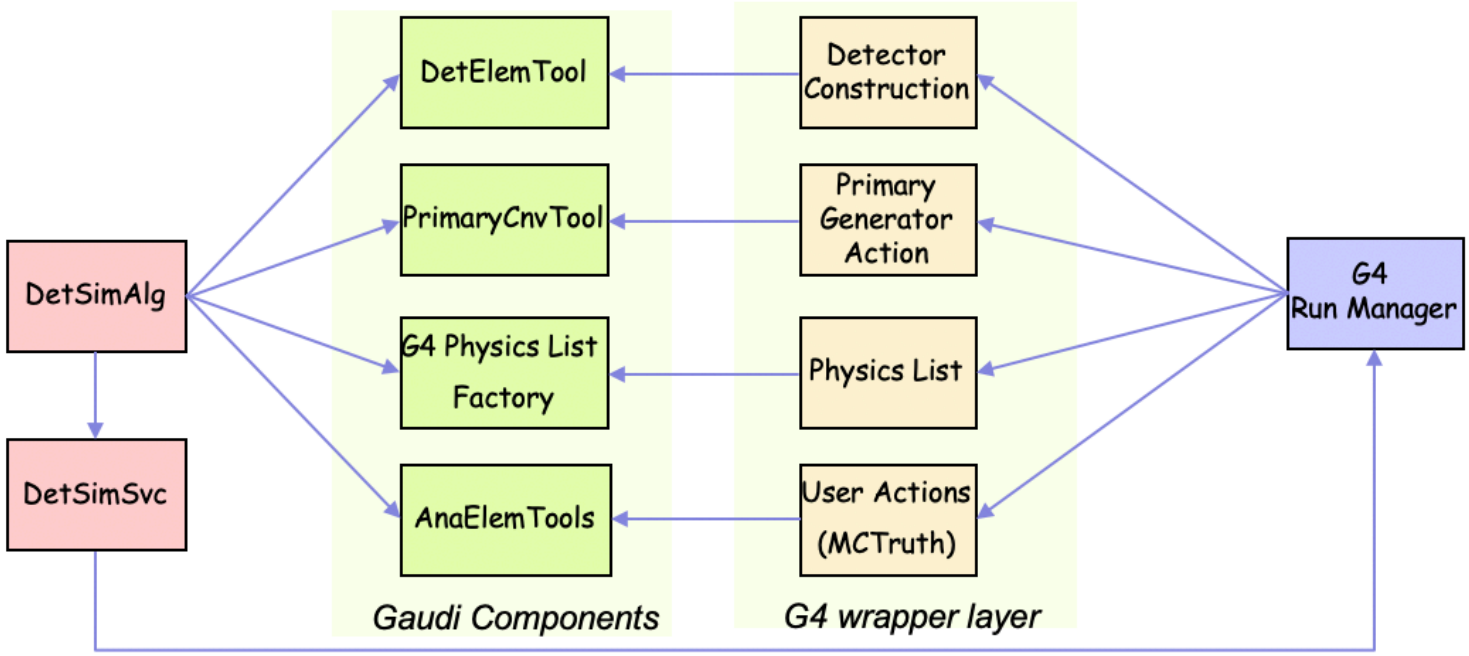
- The detector geometry convention
  - Sub-detector: both XML compact files and C++ constructor
  - the full-detector option: only compact file



- The naming conventions: DDD_oX_vYY (unique name)
  - DDD: the detector (DetCRD) or sub-detector name (ECalBarrel)
  - X: the options (hardware design)
  - YY: the implementations (software implementation)
- The different options could be easily managed in both Database and Git Repo.

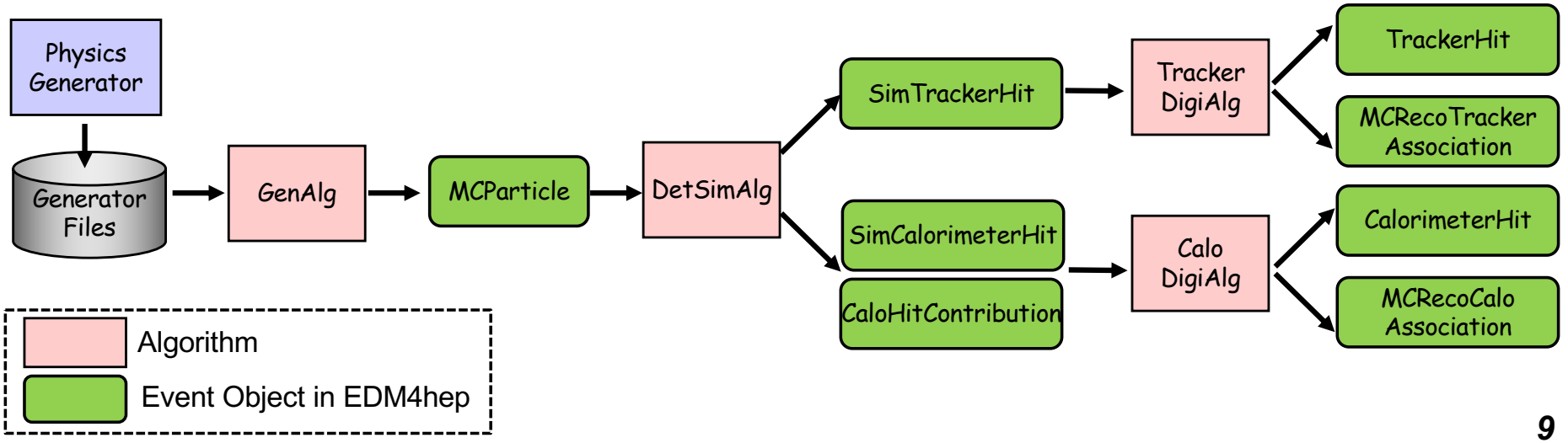
Detector Simulation Framework

- Detector Simulation Framework has been developed in CEPCSW
- Full integration with Geant4 using Gaudi components
  - Gaudi service (DetSimSvc) has a hook to G4 RunManager
  - Gaudi algorithm (DetSimAlg) invokes the DetSimSvc to simulate an event.
  - Gaudi tools are dispatched to implement detector construction, physics list and user actions of Geant4.



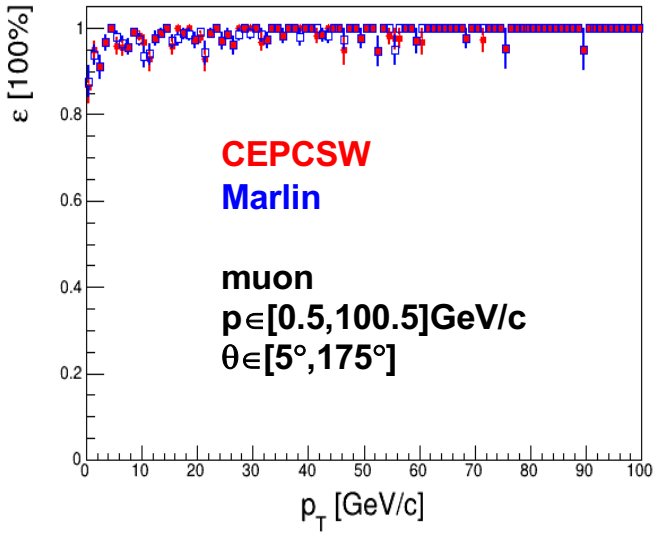
Data Associations in M.C. Data Production

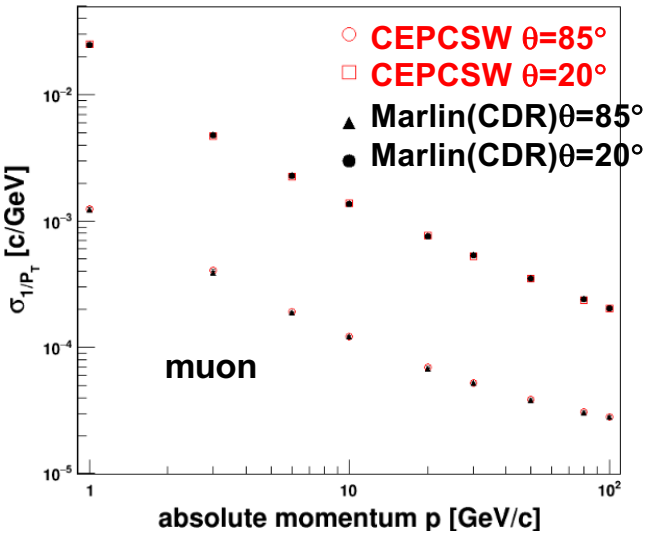
- Detector simulation provides the MCTruth information, which links the hits and primary MCParticles.
  - The Geant4 tracking action is used to build a map, which stores the relationship between primary and secondaries.
  - When creating a hit, the primary track ID will be retrieved from the map.
  - Finally, the hit object and the MCParticle is linked.
- Digitization provides the association between the Digi and the Hit via the Associations. For the real data, there is no association.



Simulation and Tracking of Tracker

- Full simulation of tracker of CEPC_v4 works well in CEPCSW
  - Detector description with DD4hep for both simulation and reconstruction.
  - Event data in the EDM4hep format (SimTrackerHit , TrackerHit)
- Finished migration of all tracking algorithms and simple tracker digitization from Marlin to CEPCSW.
- Using same LCIO simulation data, preliminary results by full tracking processor in CEPCSW are consistent with Marlin.
- Further optimization and performance study on combination of Silicon Tracker with Drift Chamber are on-going.





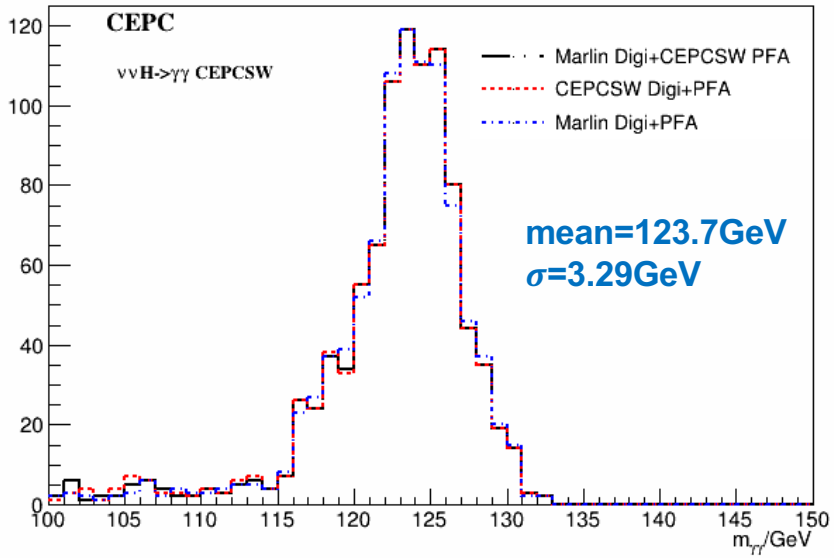
See Chengdong's talk on Dec. 28

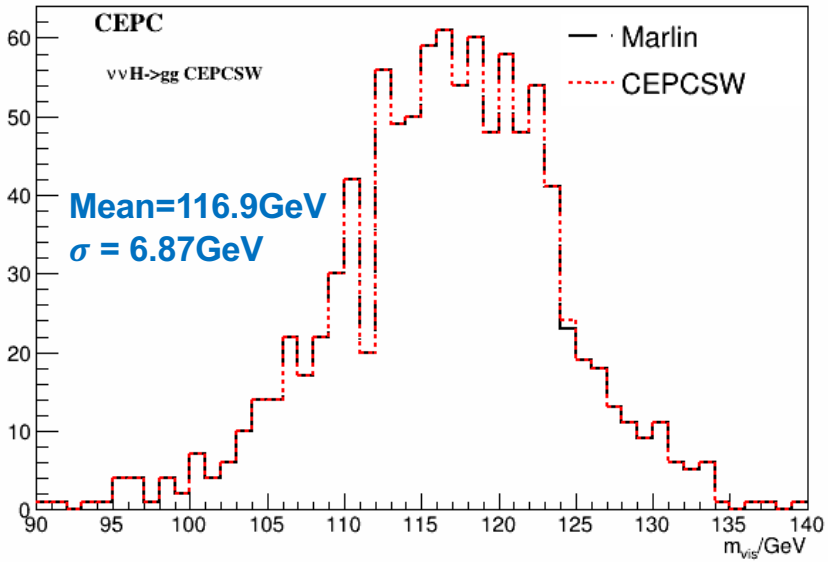
CEPC-ACTS: one tracking tool for different trackers

- Several design options of CEPC tracking system
  - Baseline (silicon + TPC)
  - Full silicon detector (FST and FST2)
  - Reference detector (silicon + drift chamber)
- Need a uniform tracking software to evaluate performance of different designs
- ACTS(A Common Tracking Software) is the most promising one
- A group is working on CEPC integration to ACTS and made good progress
- Validations of baseline design and Kalman filter have been done with truth tracking
- More details can been found in Yebo’ s talk on Dec. 28

Simulation and digitization of Calorimeter

- Full simulation of the ECAL of CEPC_v4 works well in CEPCSW
  - Detector description with DD4hep for both simulation and reconstruction.
  - Event data in the EDM4hep format (SimCalorimeterHit , CalorimeterHit)
- Calorimeter digitization has been migrated from Marlin to CEPCSW
- Validation shows the results in CEPCSW are consistent with that in Marlin





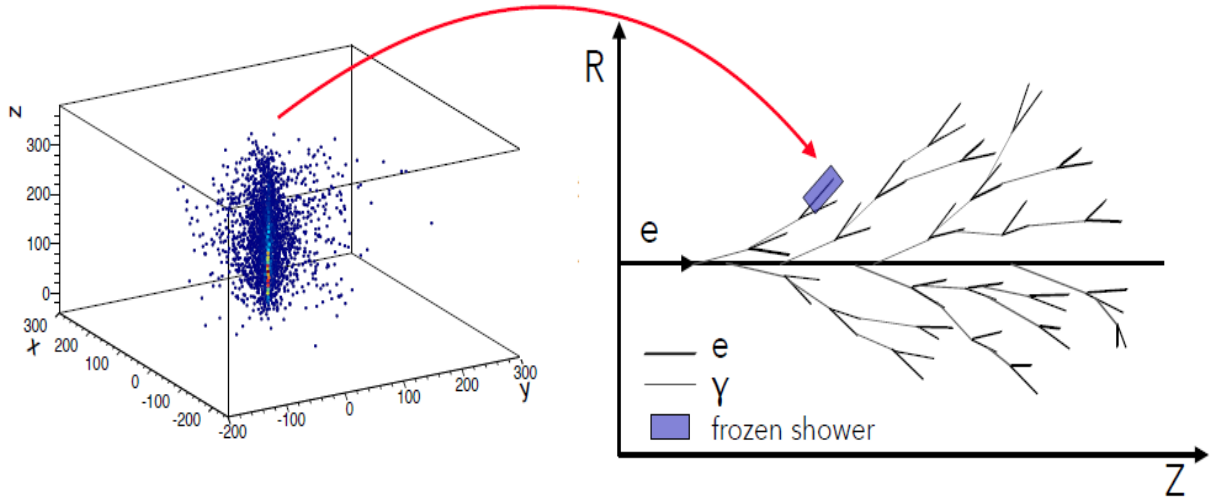
See Shengsen's talk on Dec. 28

Pandora in CEPCSW

- The majority of Higgs, W, and Z decay into multi-jets final states.
  - 3-4% jet E resolution is needed to have better 2.6-2.3 σ W/Z separation
  - Particle flow approach is the most promising to reach the goal
- Pandora is a general pattern recognition package and is used by ILC, CLIC to study PFA calorimetry.
  - Pandora App: providing input objects and receive reconstructed objects.
  - Pandora SDK: managing pandora objects.
  - Pandora Algs: reconstructing objects.
- Developed the Pandora App in CEPCSW with a Gaudi algorithm and underlyingly use Pandora SDK and Pandora Algs
- Using single γ events, the performance study shows
  - The reconstruction efficiency (above 1 GeV is > 99%)
  - The energy resolution ($\frac{16.6\%}{\sqrt{E}} \oplus 0.8\%$)
  - Both meet the CEPC CDR requirements.

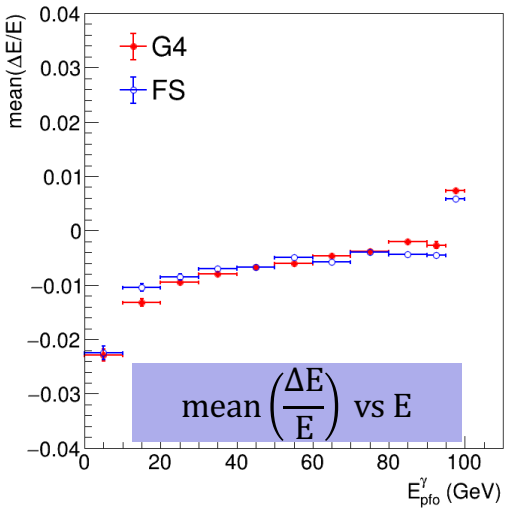
ECAL Fast Simulation: Frozen Shower

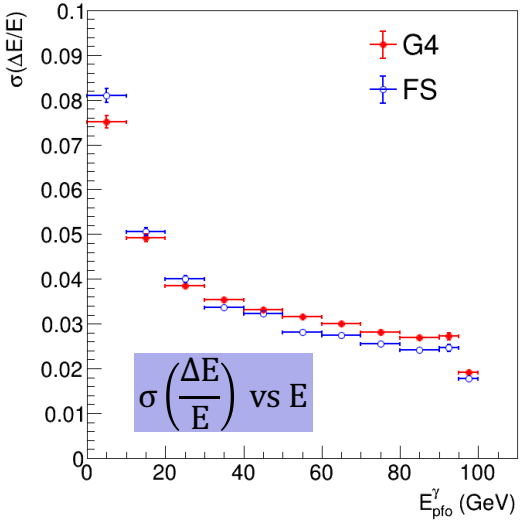
- A Full simulation of shower development costs large mount of computing and takes lots of time in Geant4.
- In Frozen shower simulation ,the low energy showers are substituted by the the pre-generated showers from the library.
- Developed ECAL fast simulation package
  - Generated Frozen Shower library of Electron/positron in barrel ECAL
  - Check reconstruction performance of γ for the FS simulation.
  - The pandora is used for reconstruction.

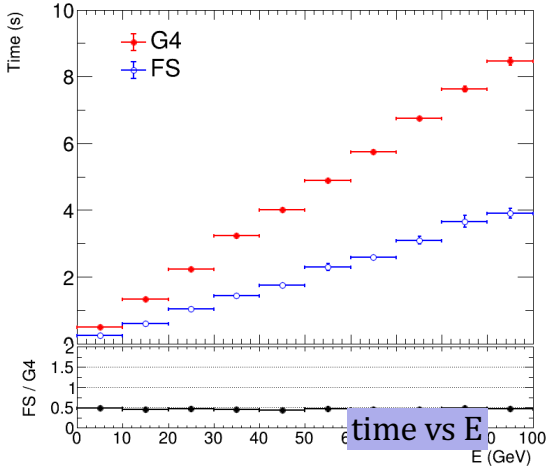


ECAL Fast Simulation: Frozen Shower

- A Full simulation of shower development costs large mount of computing and takes lots of time in Geant4.
- In Frozen shower simulation ,the low energy showers are substituted by the the pre-generated showers from the library.
- Developed ECAL fast simulation package
  - Generated Frozen Shower library of Electron/positron in barrel ECAL
  - Check reconstruction performance of γ for the FS simulation.
  - The pandora is used for reconstruction.



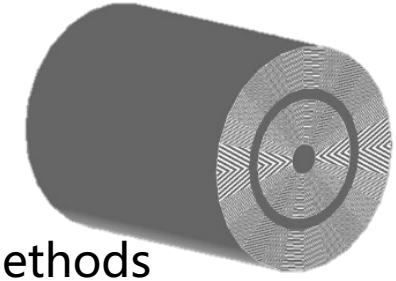


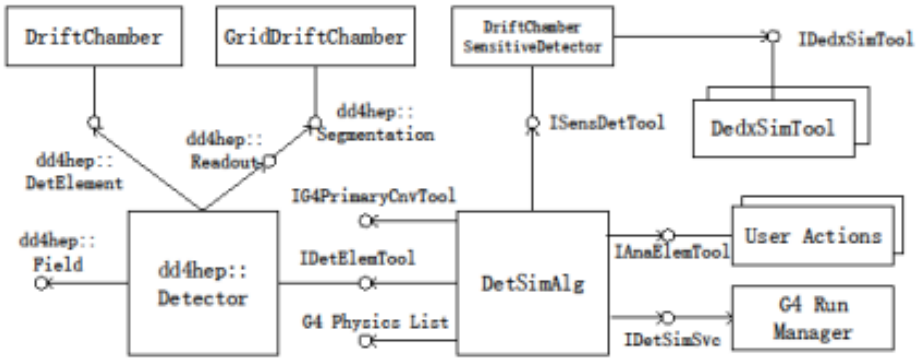


Reference Detector Study: Drift Chamber

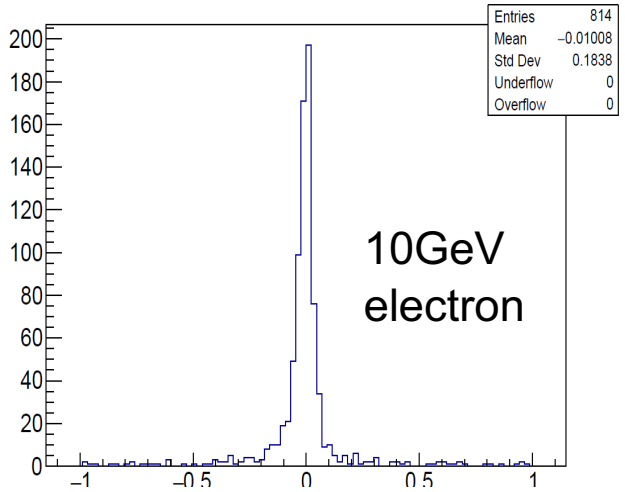
- Two drift chambers implemented in CEPCSW
  - Baseline:1.8m, 130 layers, He:iC_4H_{10}=90:10
  - Cell partitioning with the segmentation method
- dE/dx simulation with sampling method
  - Configurable Gaudi tools could be used for different dE/dx methods
- Track fitting with Kalman filtering use Genfit2
  - Space points which random selected from truth as input
- Will focus on Simulation of waveform for cluster counting method, Development of tracking algorithms and performance studies

Geometry of CEPC drift chambers





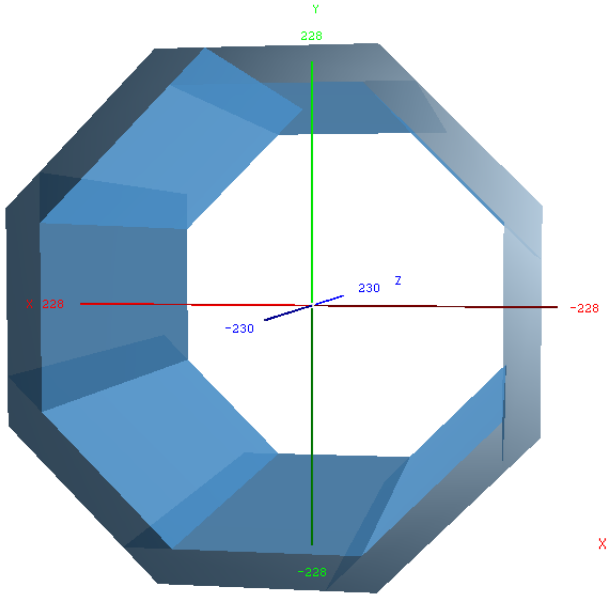
UML of simulation framework

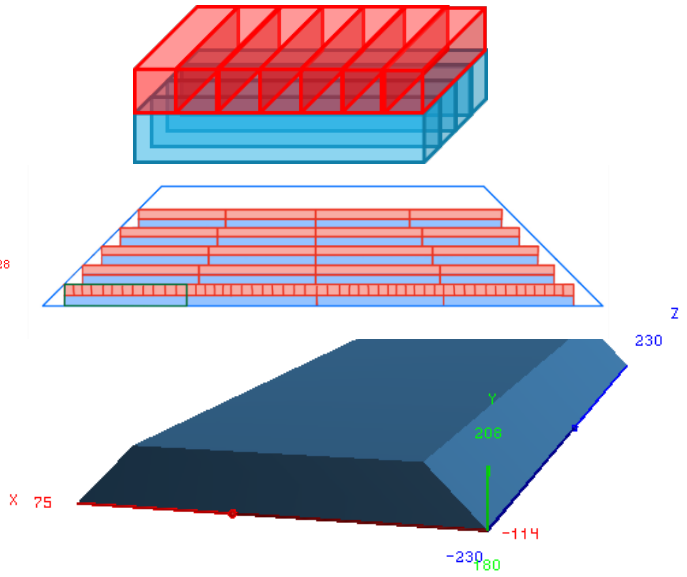


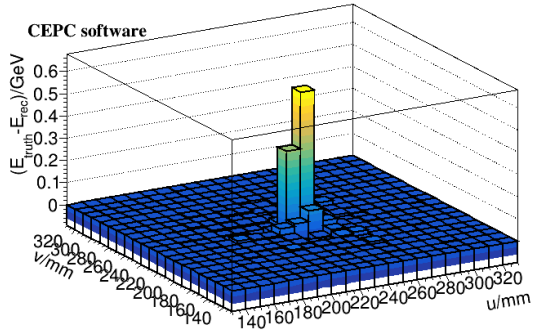
Kalman filter momentum

Reference Detector Study: Crystal Bar ECAL

- A crystal bar ECAL design provides optimal intrinsic energy resolution and reduction of number of channels.
- The crystal bar ECAL has been implemented in CEPCSW, and simplified/parameterized simulation and digitization are ready.
- Preliminary results show that 3D showers profiles can be extracted from 2D measurements of a single high-energy γ.
- Major focus in the plan: multiple incident particles hitting one super cell (e.g. ambiguity, energy splitting).







Software Environment

- CEPCSW uses the modern C++ 17 standard as default.
- It supports both LCG stack and Key4hep stack, whose pre-compiled external libraries are distributed via CVMFS.
  - LCG stack: LCG 97 + CEPCSW specific libraries,
    - /cvmfs/cepcsw.ihep.ac.cn/prototype/releases/externals/97.0.2/setup.sh
  - Key4hep stack: managed by spack. CEPCSW could be also installed by spack.
    - /cvmfs/sw.hsf.org/key4hep/setup.sh
- CEPCSW has been deployed via CVMFS and is accessible in all CEPC sites
  - Setup environment via CVMFS: source setup.sh under CEPCSW
  - install CEPC software via Spack: spack install cepcsw
- For personnel PC or other non-CEPC sites, two steps of CVMFS configuration are needed
  - Get IHEP key and put in /etc/cvmfs/keys/ihep.ac.cn
  - Put server info in /etc/cvmfs/domain.d/ihep.ac.cn.conf

Documentation and Development Workflow

- CEPCSW is managed with GitHub and are open source!
- Documentation:
  - CEPCSW Installation: https://github.com/cepc/CEPCSW/blob/master/.travis.yml#L15
  - Quick Start: https://github.com/cepc/CEPCSW/blob/master/docs/quickstart.md
  - CEPCSW Tutorial organized in Sep. 2020: https://indico.ihep.ac.cn/event/12341/
  - EDM4hep: https://github.com/key4hep/EDM4hep
  - DD4hep: https://github.com/AIDASoft/DD4hep
- The development workflow:
  - Create an issue (https://github.com/cepc/CEPCSW/issues)
  - Create a pull request (https://github.com/cepc/CEPCSW/pulls)
    - Fork the official repo and make changes in developer repo
  - Travis CI (Continues Integration) runs the build and test automatically
    - https://travis-ci.com/github/cepc/CEPCSW
  - Review and merge
  - Close the issue

Summary & Plan

- Lots of progress has been made since last CEPC workshop
  - CEPCSW is fully integrated with Key4hep and extensively used for applications
  - Event Data Model has been moved to EDM4hep
  - Detector simulation framework is developed and used for the study of CEPC_v4 detector and reference detector
  - ECAL fast simulation with the frozen shower method is developed to speed up the simulation of electromagnetic shower.
  - Finished porting of digitization and reconstruction algorithms for trackers and ECAL from Marlin to CEPCSW within schedule
  - The k4Pandora package is developed to integrate Pandora with CEPCSW and became part of Key4hep software stack
- CEPCSW is managed with Github, deployed with CVMFS, and available for all CEPC Sites

Summary & Plan

- Plan
  - Adding more components from Key4hep when they are available
  - Non-uniform magnetic field and piling-up of beam backgrounds
  - Development of simulation and reconstruction algorithms for the reference detector （SiTrk+DC, Crystal bar ECal)
  - Adding algorithms for building reconstructed particles
  - Checking the consistence of software with benchmark performance studies.

Welcome More Collaborators to Join in the Software Group !

Thank You !

謝謝

Backup

Development environment

- The development environment could be setup via CVMFS or Spack.
- As all the external libraries are distributed via CVMFS, we encourage people to use them directly, which could avoid the installation from scratch.
  - Install the CVMFS client in the host
  - Use the docker image provided by CEPCSW: **docker pull cepc/cepcsw**
  - The Dockerfile is also available in the CEPCSW repo: https://github.com/cepc/CEPCSW/blob/master/Dockerfile
- The installation via spack is also available in the k4-spack repo
  - https://github.com/key4hep/k4-spack
  - The spack receipt of CEPCSW: https://github.com/key4hep/k4-spack/blob/master/packages/cepcsw/package.py

Access CEPC software on CVMFS

- CEPC software has been deployed globally via CVMFS
  - Stratum0 server in IHEP and two Stratum1 servers in IHEP and RAL
  - CEPC software can be found in /cvmfs/cepc.ihep.ac.cn with CVMFS client installed and configured
  - Now it is accessible in all CEPC sites
- For personnel PC or other non-CEPC sites, two steps of CVMFS configuration to get access
  - Get IHEP key and put in /etc/cvmfs/keys/ihep.ac.cn
    - http://cvmfs-stratum-one.ihep.ac.cn/cvmfs/software/client_configure/ihep.ac.cn/ihep.ac.cn.pub
  - Put server info in /etc/cvmfs/domain.d/ihep.ac.cn.conf
    - CVMFS_SERVER_URL="http://cvmfs-stratum one.ihep.ac.cn:8000/cvmfs/@fqrn@;http://cvmfs-egi.gridpp.rl.ac.uk:8000/cvmfs/@fqrn@"

Site	SiteType
GRID.QMUL.uk	GRID
CLUSTER.SJTU.cn	CLUSTER
CLUSTER.IPAS.tw	CLUSTER
GRID.LANCASTER.uk	GRID
CLOUD.IHEPCLOUD.cn	CLOUD
GRID.IHEP.cn	GRID