Status of the CEPCSW Prototype

Jiaheng Zou

On behalf of CEPCSW working group

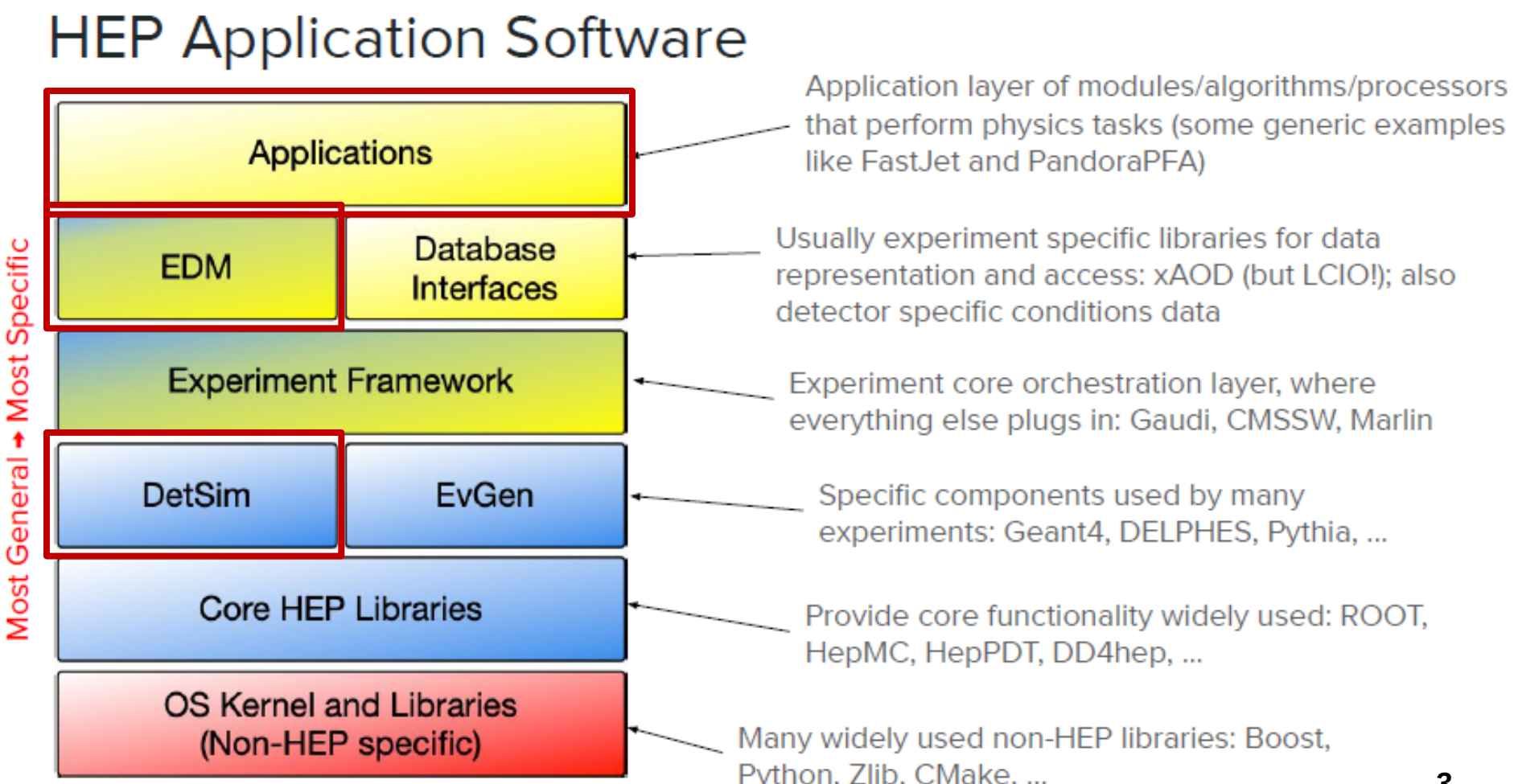
2019.11.01

The Goal of CEPCSW Prototype

- Based on CSS (Common Software Stack)
- Reuse existing components
  - DD4hep, Gaudi, ROOT …
- Implement the specific components for CEPC
- Provide a ready-to-work environment to algorithm developers and physicists
  - Migrate marlin algorithms to CEPCSW
  - Integrate more algorithms and features
- Move from marlin to the new software system

Common Software Stack (CSS)

A common solution for future collider experiments: iLC, FCC, CEPC



Tasks

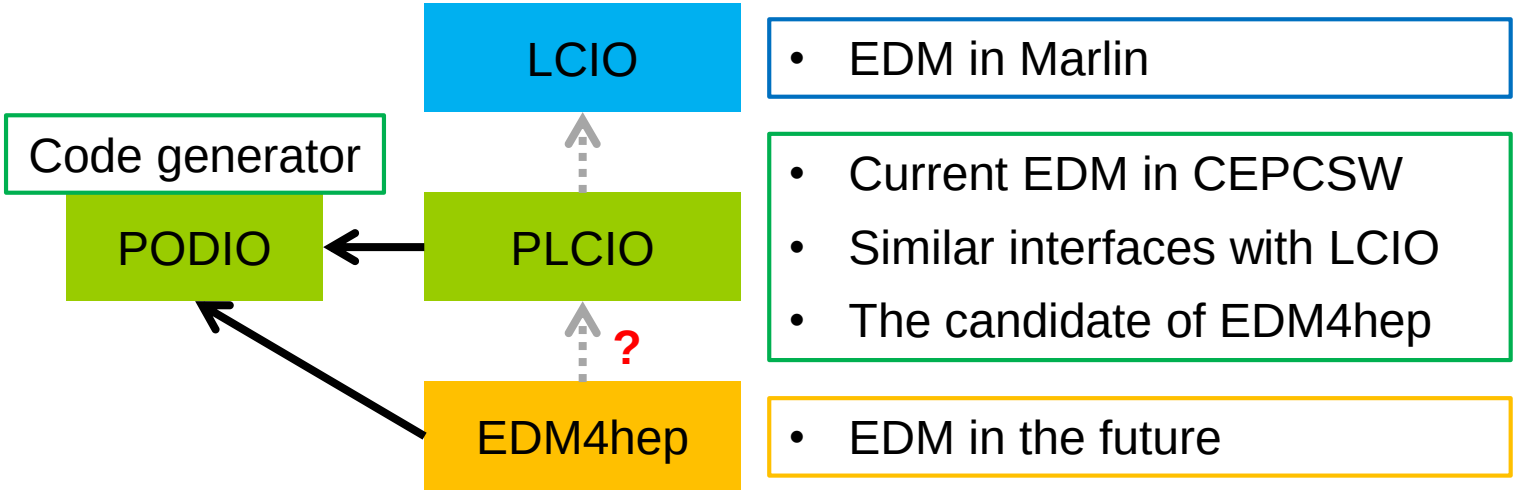
- Event Data Model
- Simulation
- Application layer modules (Reconstruction)
- Integration and testing

Event Data Model : EDM4hep

- The EDM4hep project is being constructed in the context of CSS
  - Based on PODIO, or plain-old-data I/O
    - A code generator – generate classes from yaml files
    - Avoid deep-object hierarchies and virtual inheritance
    - Improve runtime performance
    - Simplify the implementation of data reading/writing
  - Common core classes described in a yaml file
  - Each experiment can implement their own extensions
  - A project followed by HEP Software Foundation
    - Regular meeting in every 2 weeks (CERN, DESY, IHEP …)
    - https://github.com/HSF/EDM4hep
  - But, it is not ready yet : (

Current EDM in CEPCSW Prototype

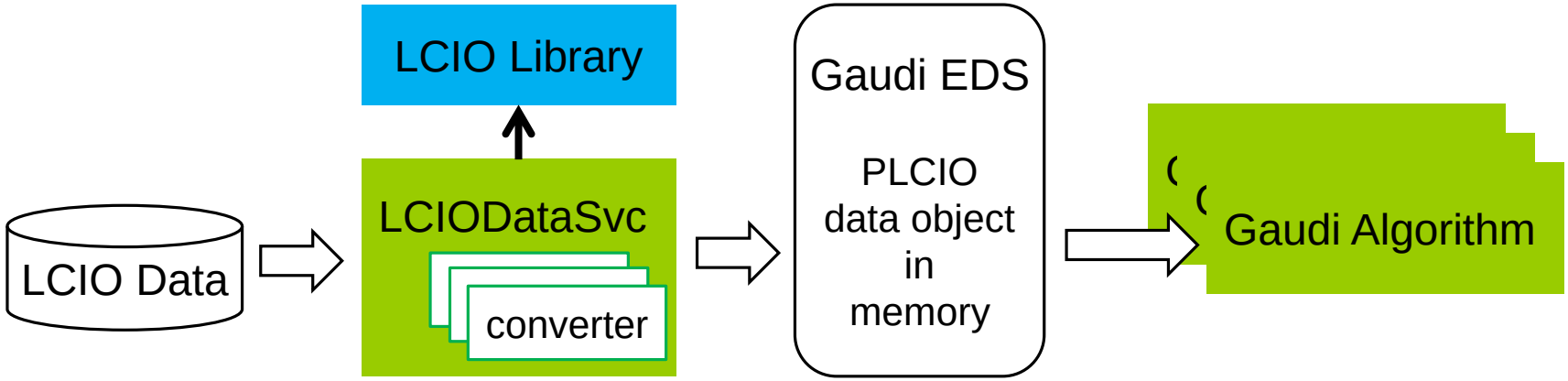
- CEPCSW will use PLCIO before EDM4hep is ready
- PLCIO is an implementation of the LCIO event data model in PODIO



- Possibly EDM4hep will be inherited from PLCIO
  - The migration from PLCIO to EDM4hep should be easy
- We are the first user of PLCIO
  - Some missing classes – implement by ourselves
  - Potential problems, such as memory leak – need more debugging

Read the Existing LCIO Data

- LCIODataSvc
  - Read LCIO files via the LCIO library
  - Convert LCIO data objects to PLCIO data objects
  - Register PLCIO data objects to Gaudi Event Data Store



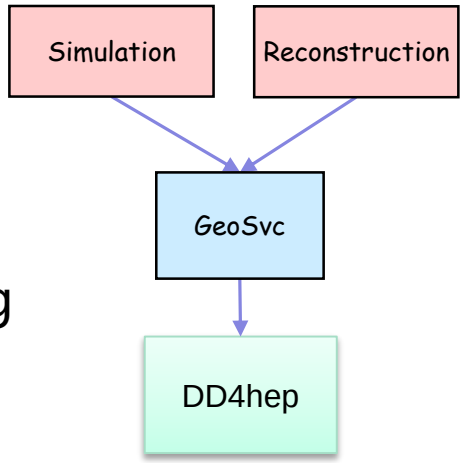
- Current Status
  - Data converters for DST data types
  - Some of the data relations are not recovered properly yet

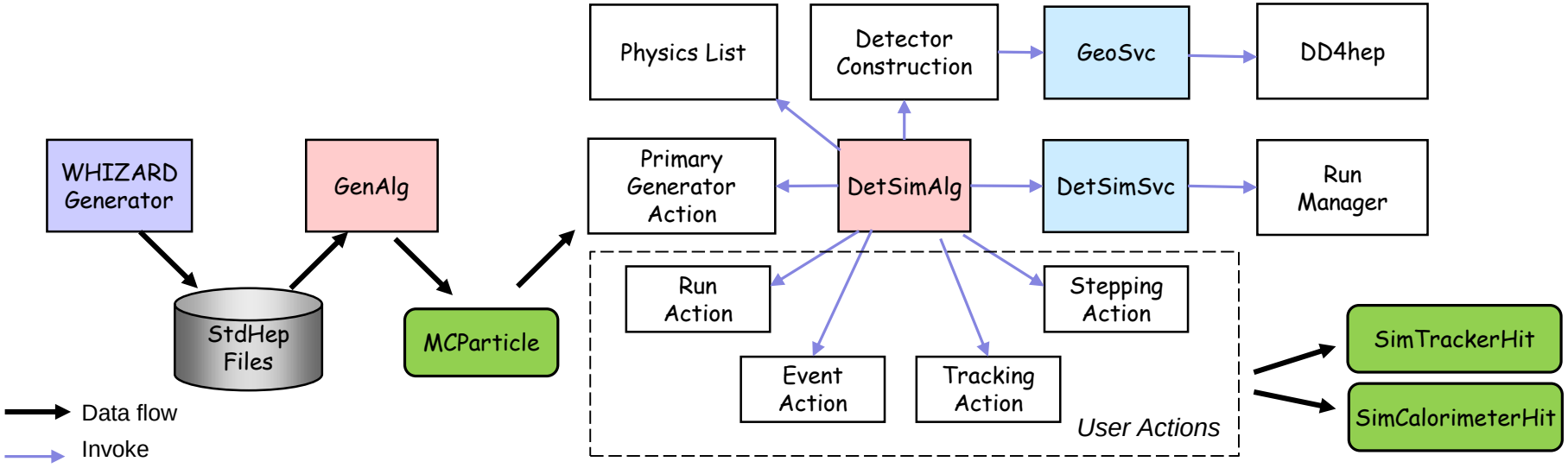
Tasks

- Event Data Model
- Simulation
- Application layer modules (Reconstruction)
- Integration and testing

Status of Simulation Framework (I)

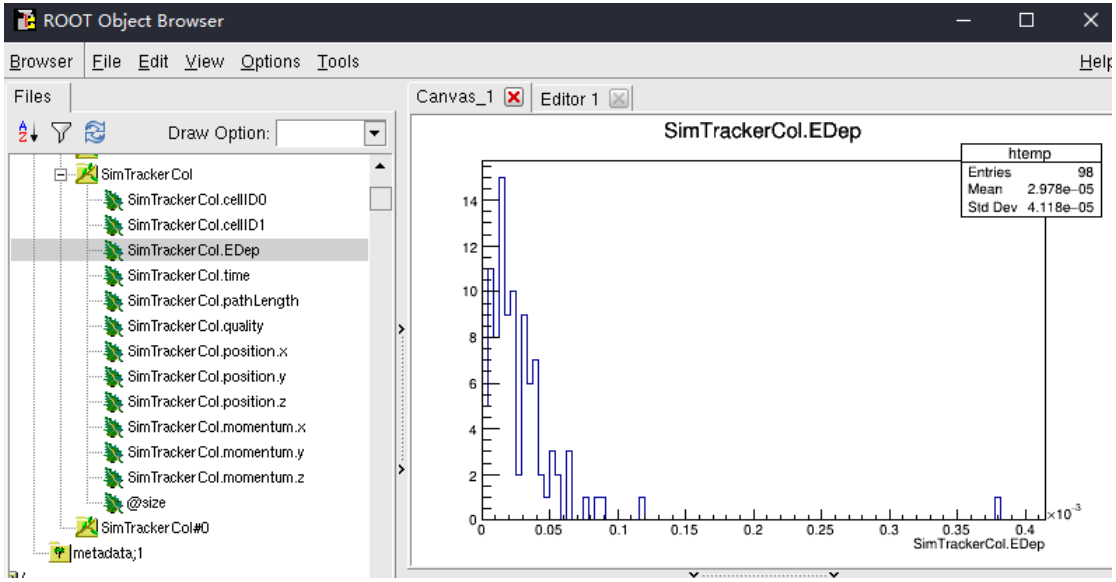
- Integration with Gaudi & DD4hep is done
  - Geometry:
    - CEPC_v4 (DD4hep version) from Chengdong
    - Silicon Detector is enabled
  - The main processing procedure of the simulation framework





Status of Simulation Framework (II)

- Execution test
  - Input: StdHep and LCIO formats
  - Output: plcio, converting from the DDG4 Hit objects (Tracker/Calorimeter)



- Next Steps
  - integrate with digitization algorithm
  - validation with existing MC samples

Tasks

- Event Data Model
- Simulation
- Application layer modules (Reconstruction)
- Integration and testing

SiliconTracking Migration

- Pixel: SimTrackerHit → TrackerHit → SiliconTracking
- Strip: SimTrackerHit → TrackerHit → SpacePoint → SiliconTracking
- Tracking processes:
  - SiliconTracking_MarlinTrk —
  - ↓
  - TrackSubsetProcessor→ ~~FullLDCTracking_MarlinTrk~~
  - ↑
  - ForwardTracking —
  - ~~ClupatraProcessor~~ —↑
- Package dependencies → module package (by module classes)
  - MarlinTrk
  - KalDet
  - KalTest
- Data model dependencies, LCIO to plcio
    - EVENT::TrackerHit → plcio::TrackerHit
    - EVENT::TrackerHitPlane → plcio::TrackerHitPlane

Progress

package	Component of CEPCSW	Type	status
MarlinTrk	TrackSystemSvc	service	Compile & link
KalDet	KalDet	Independent	Compile & link
KalTest	KalTest	Independent	Compile & link
SpacePointBuilder	SpacePointAlg	Algorithm	ongoing
SiliconTracking_MarlinTrk	SiliconTrackingAlg	Algorithm	ongoing
ForwardTracking	?	?	?
TrackSubsetProcessor	?	?	?

- After implementing module packages, similar usage for fitting

Issues and Solutions (Temporary)

- Can not get relative object directly in plcio data model
  - Remove relative object usage in tracking and after fitting, influence some following analysis based on MC truth particles
  - We are going to extend plcio with a service to handle the object relations
- Can not convert pointers of data objects each other
  - Fix data type in algorithm and use template in codes of module classes
- Geometry service during the transition from Marlin to Gaudi
  - Use the GEAR package in Marlin at present
  - Will be replaced by the new service with DD4hep

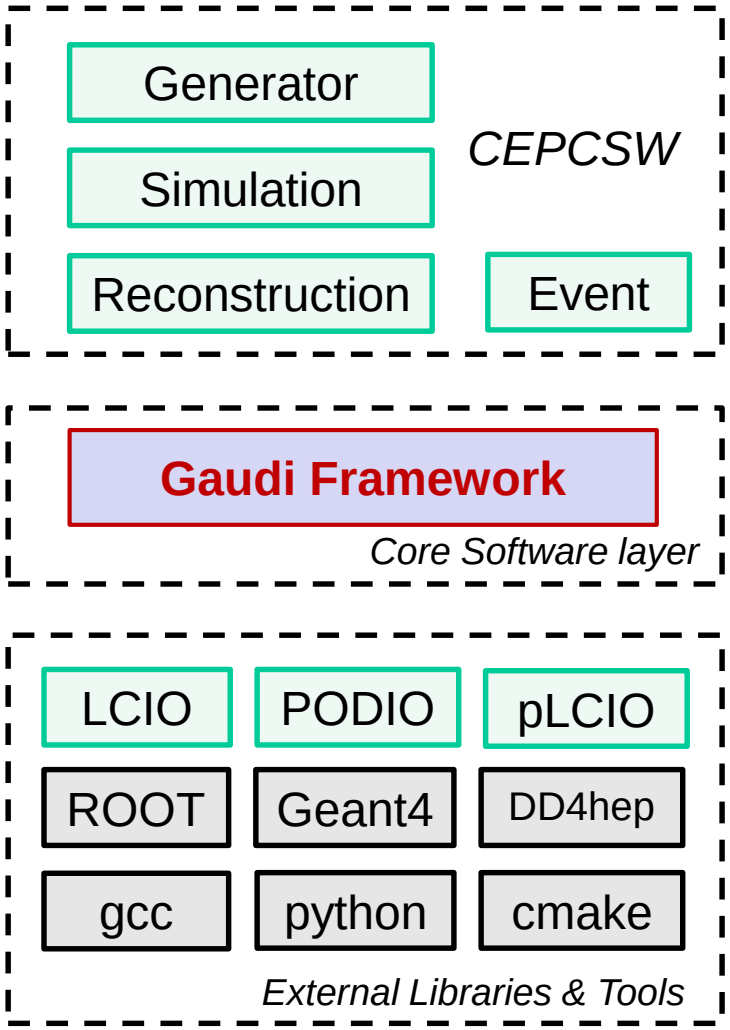
Tasks

- Event Data Model
- Simulation
- Application layer modules (Reconstruction)
- Integration and testing

Software Infrastructure and Building

- Common tools
  - CMake: Build & deployment
    - Gaudi cmake macros
  - Git: version control
    - http://cepcgit.ihep.ac.cn/cepc-prototype
  - CVMFS: software distribution
    - CEPC specific:

    /cvmfs/cepcsw.ihep.ac.cn/prototype
- Software building
  - Based on LCG software stack now
  - Move to KEY4hep in the context of CSS in the future



A Preliminary Testing

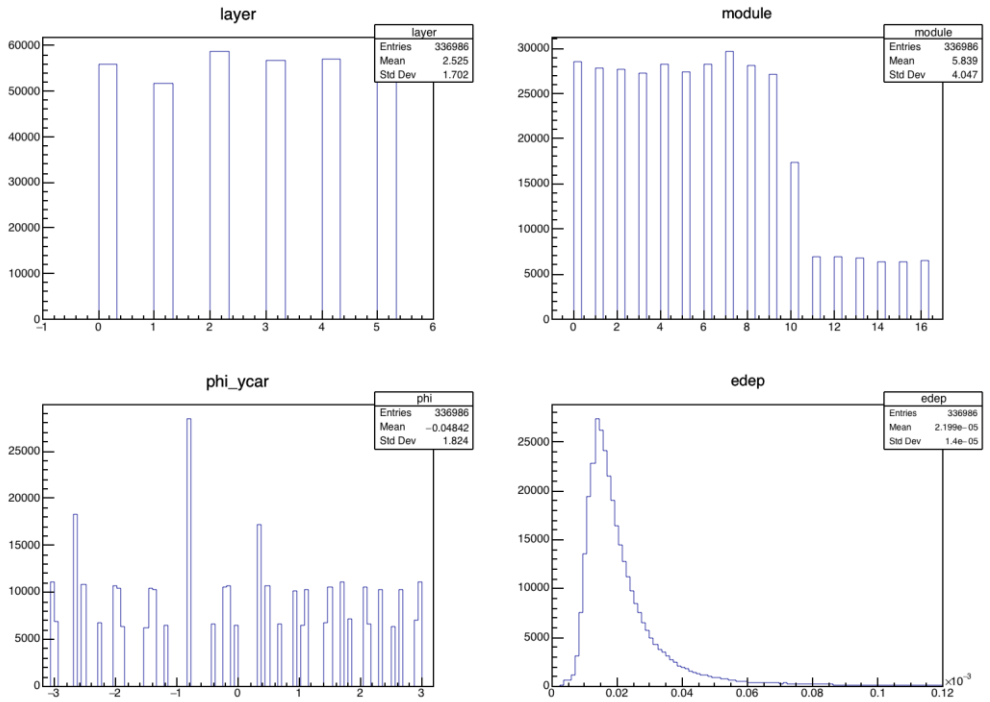
- A digitization algorithm migrated from Marlin
- Geometry: GearSvc migrated from Marlin
- Data and I/O
  - Read .slcio (LCIO) format files with LCIODataSvc
  - Write .podio (PLCIO) format files with PodioDataSvc
- Compare the results with Marlin

Physics Results

- The results of CEPCSW and Marlin are exactly the same

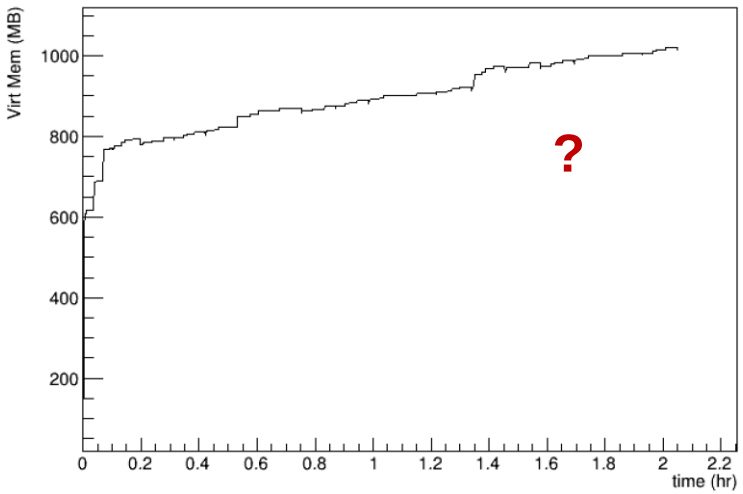
```
Attaching file PlanarDigi_marlin.root
(TFile *) 0x7fc1ef93ea70
root [1] planarDigi->Show(13000)
======> EVENT:13000
 side            = 0
 layer           = 3
 module          = 8
 sensor          = 0
 theta_xcar      = 1.5708
 phi_xcar        = 1.428
 theta_ycar      = 8.65927e-17
 phi_ycar        = -2.49899
 edep            = 4.89194e-05
```

```
Attaching file PlanarDigi_gaudi.root
(TFile *) 0x7f9b39d043d0
root [1] planarDigi->Show(13000)
======> EVENT:13000
 side            = 0
 layer           = 3
 module          = 8
 sensor          = 0
 theta_xcar      = 1.5708
 phi_xcar        = 1.428
 theta_ycar      = 8.65927e-17
 phi_ycar        = -2.49899
 edep            = 4.89194e-05
```

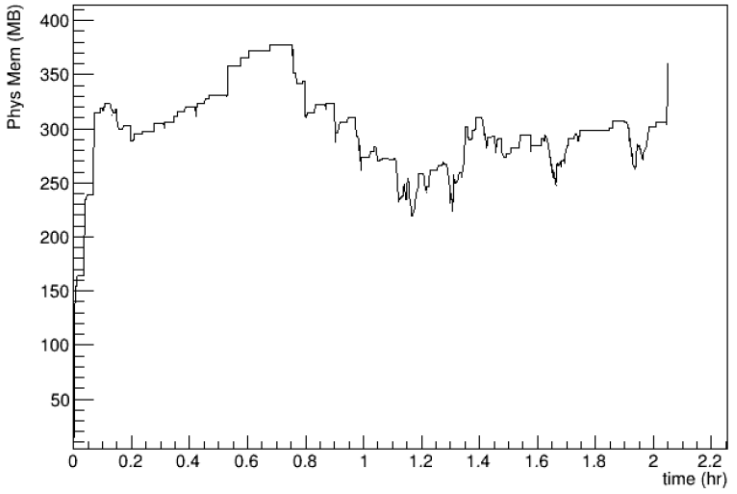


Execution Performance

- Execution time
  - CEPCSW – 124 min VS. Marlin – 131 min
- Memory usage of CEPCSW



Virtual memory



Physical memory

Summary

- The CEPCSW prototype begins to take shape
  - EDM, I/O and common services
  - Simulation framework and DD4hep integration
  - Reconstruction algorithms
  - A reasonable testing result
- Have joined the international collaboration on Common Software Stack for future HEP experiments
  - EDM4hep, KEY4hep …
- Plans
  - Release a first workable demo before the Nov. workshop
  - Migrate more algorithms from Marlin
  - High performance computing development