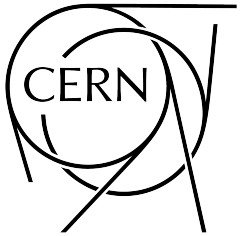


CERN Summer Student Programme 2024

Porting and validation of reconstruction algorithms to the Key4hep framework

Katerina Kostova

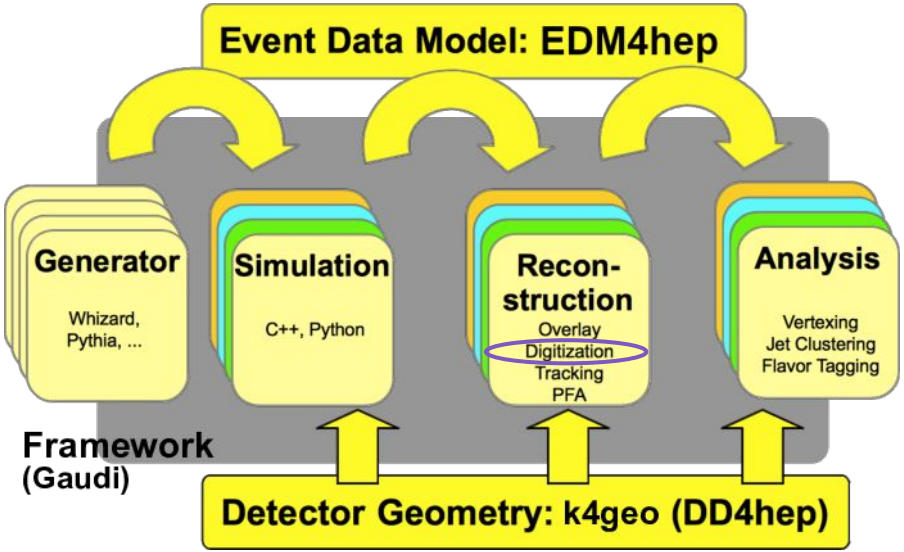
Supervisors: Juan Miguel Carceller, Swathi Sasikumar

23 September 2024

Introduction

What is Key4hep?

- turnkey software for future colliders that provides full experiment life-cycle
- contributions and usage from CLIC, ILC, FCC, Muon collider, etc. communities



Key4hep main components:

- Gaudi as the processing framework
- DD4hep for the geometry information
- EDM4hep for the event data model

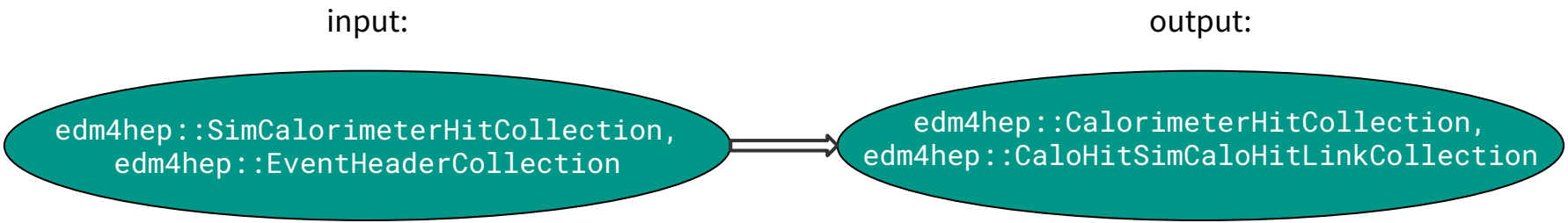
My project:

- Reconstruction algorithms

Introduction

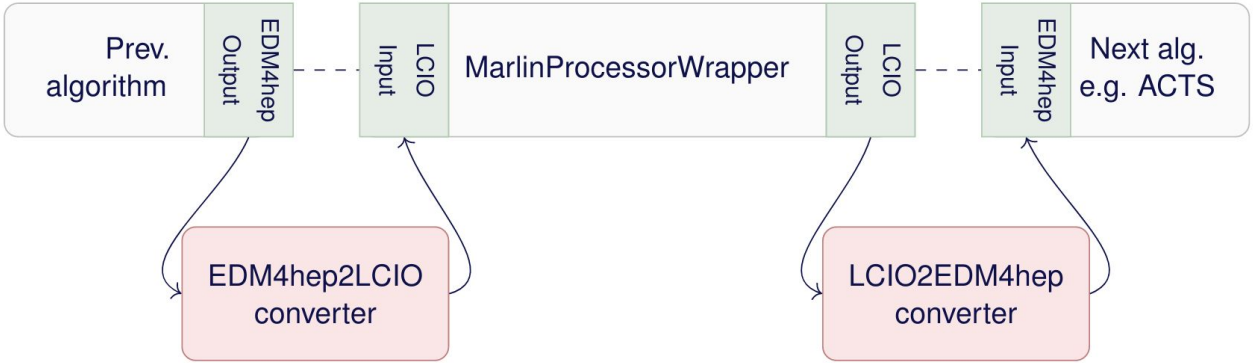
What is a digitisation algorithm?

- part of the reconstruction process - prepare the data for physical analysis
- transform the “raw data” generated by the simulation into a digital signal
- correct detector effects
  - calibration
  - thresholds
  - time corrections etc.
- in Key4hep: `edm4hep::k4FWCore::MultiTransformer` algorithm is used



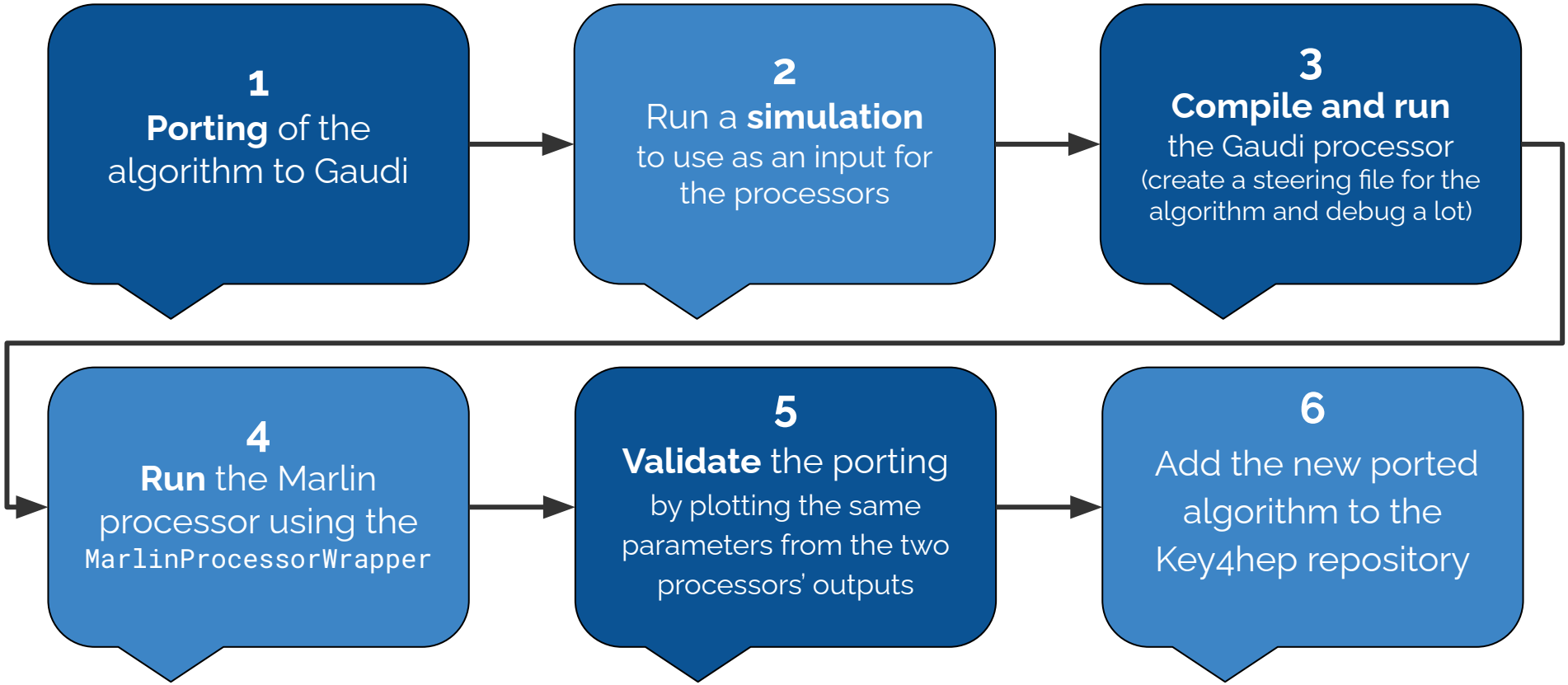
Motivation

1. *Integration* of iLCSoft algorithms (used by ILC and CLIC) from Marlin framework to Gaudi to be used in Key4hep
   - currently k4MarlinWrapper is used to run Marlin processors in Gaudi



2. *Validation* as an important step in porting algorithms into new framework

Workflow

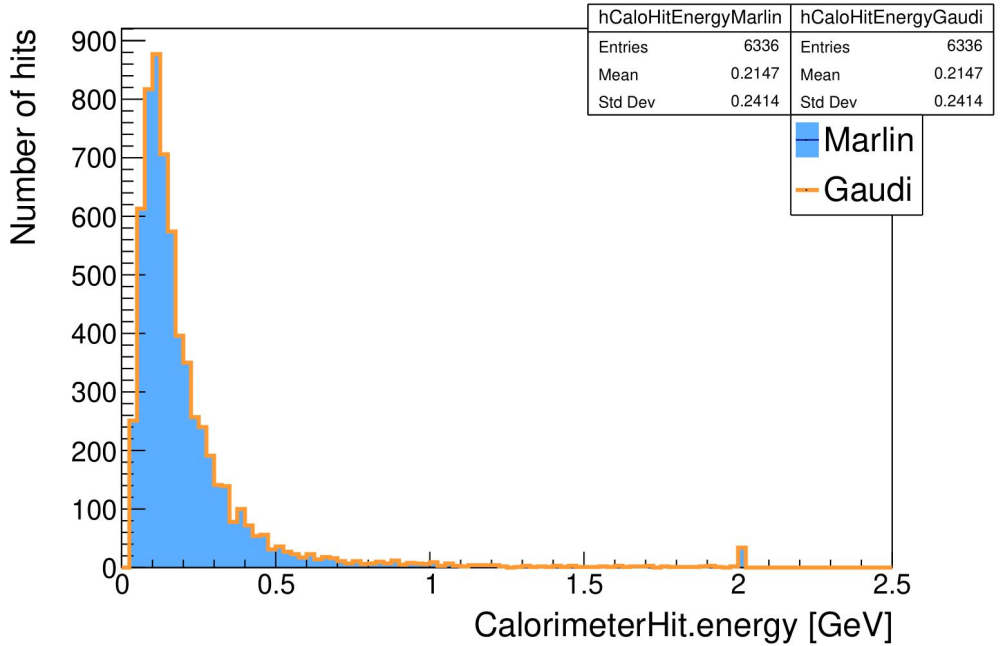

1
Porting of the algorithm to Gaudi
2
Run a **simulation** to use as an input for the processors
3
Compile and run the Gaudi processor (create a steering file for the algorithm and debug a lot)
4
Run the Marlin processor using the `MarlinProcessorWrapper`
5
Validate the porting by plotting the same parameters from the two processors' outputs
6
Add the new ported algorithm to the Key4hep repository

DDSimpleMuonDigi

DDSimpleMuonDigi

- simulation: 1000 events - muon particle gun with 10 GeV energy - CLD detector
- use YokeBarrel and YokeEndcap collections

→ compare CalorimeterHit.energy for Marlin and Gaudi processors

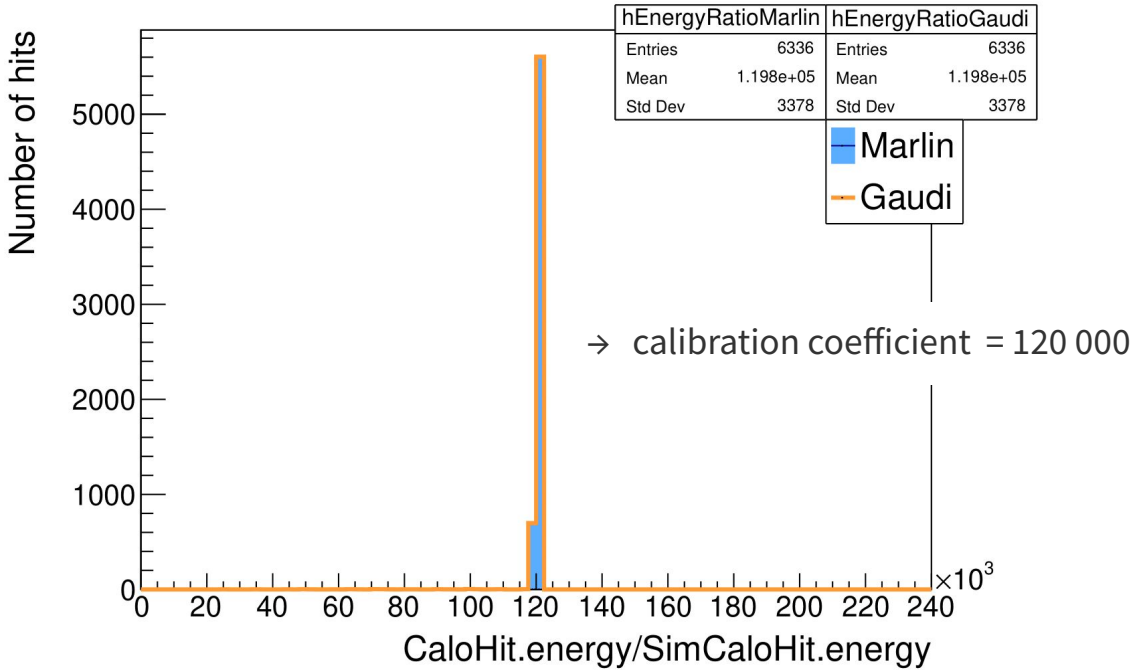


What is the **DDSimpleMuonDigi** algorithm doing?

- multiply the energy of every *SimCalorimeterHit* by a **calibration coefficient** (120 000)
- if *CalorimeterHit.energy* > **max hit energy** (2.0 GeV), then write *CaloHit* energy = max hit energy
- if *CalorimeterHit.energy* > **energy threshold** (0.025 GeV), then save all info about the hit

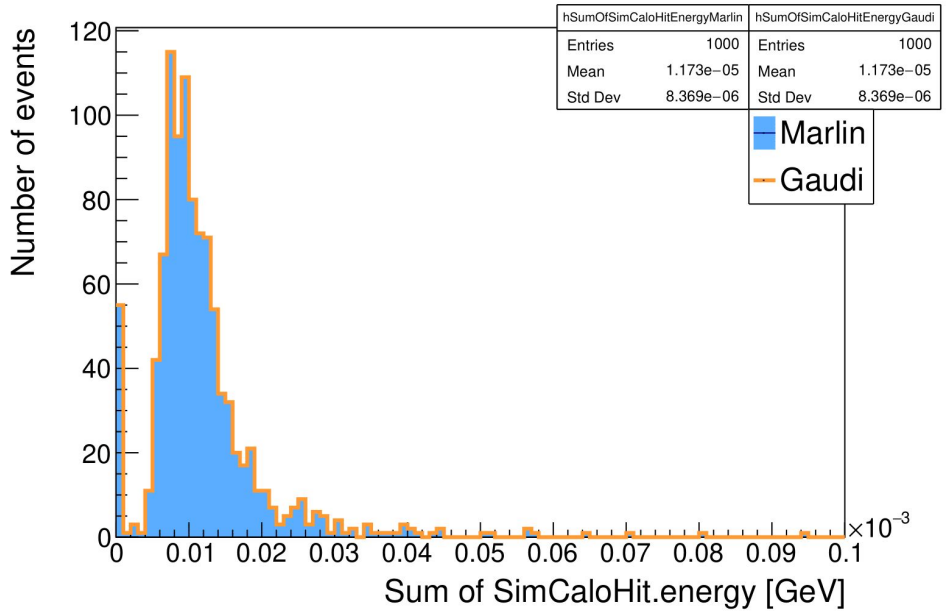
DDSimpleMuonDigi

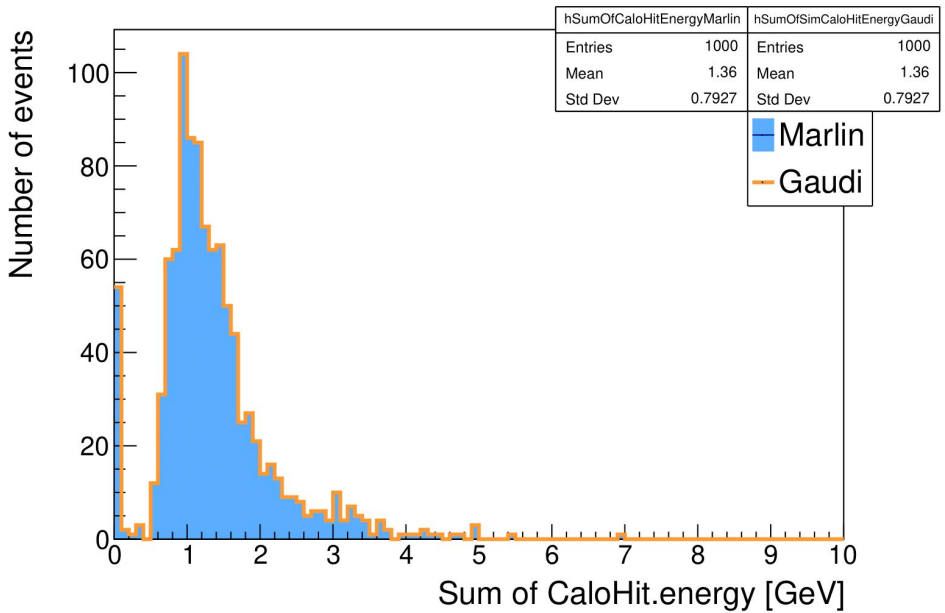
→ compare the ratio between *CalorimeterHit.energy* and *SimCalorimeterHit.energy* for every reconstructed hit (use `edm4hep::CaloHitSimCaloHitLinkCollection`)



DDSimpleMuonDigi

→ compare sum of *SimCalorimeterHit.energy* per event and sum of *CalorimeterHit.energy* per event





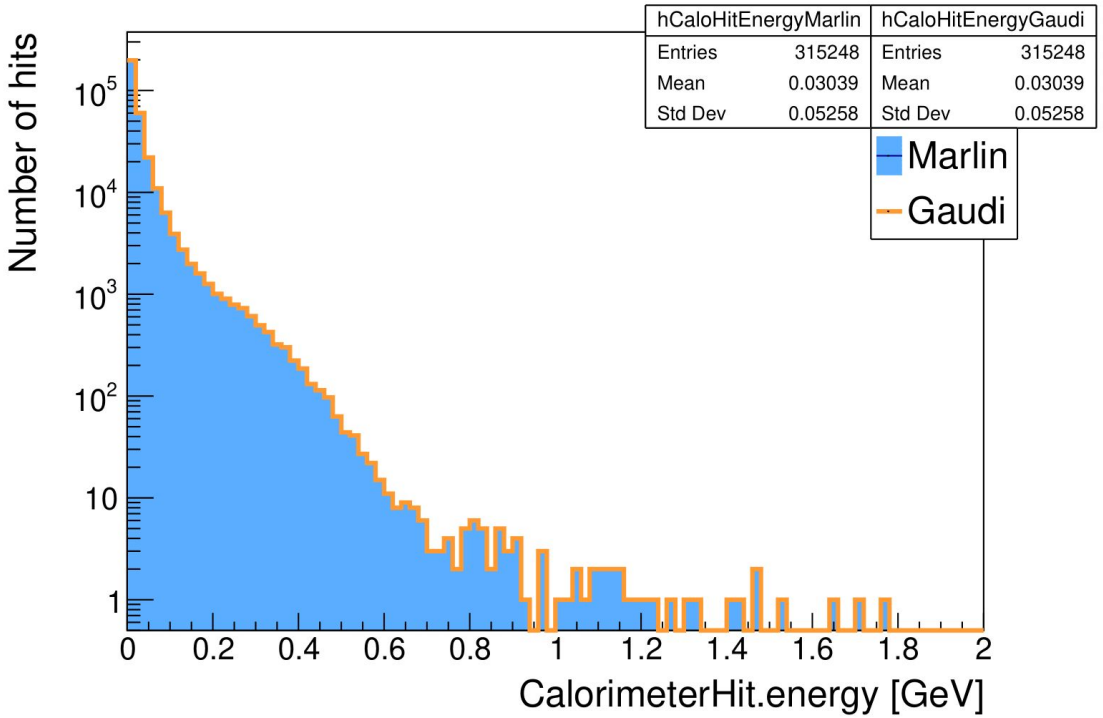
→ sum of *CalorimeterHit.energy* per event don't match the simulated particle energy (10 GeV) - muons are not fully absorbed in the muon systems

DDCaloDigi

DDCaloDigi

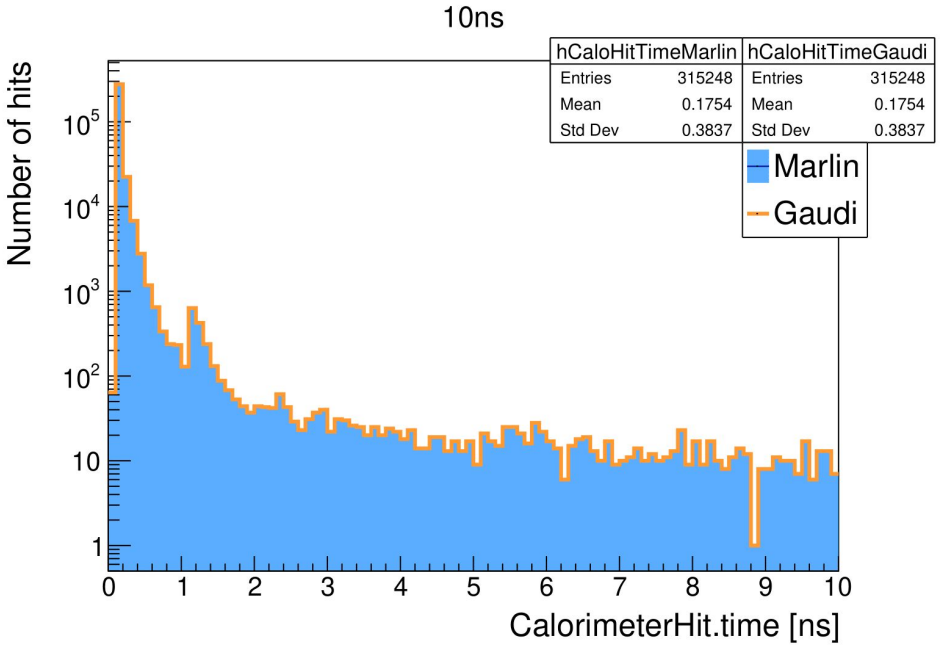
- simulation: 1000 events - photon particle gun with 10 GeV energy - CLD detector
- use collections:
  - ECALBarrel
  - ECALEndcap
  - HCALBarrel
  - HCALEndcap
  - HCALOther

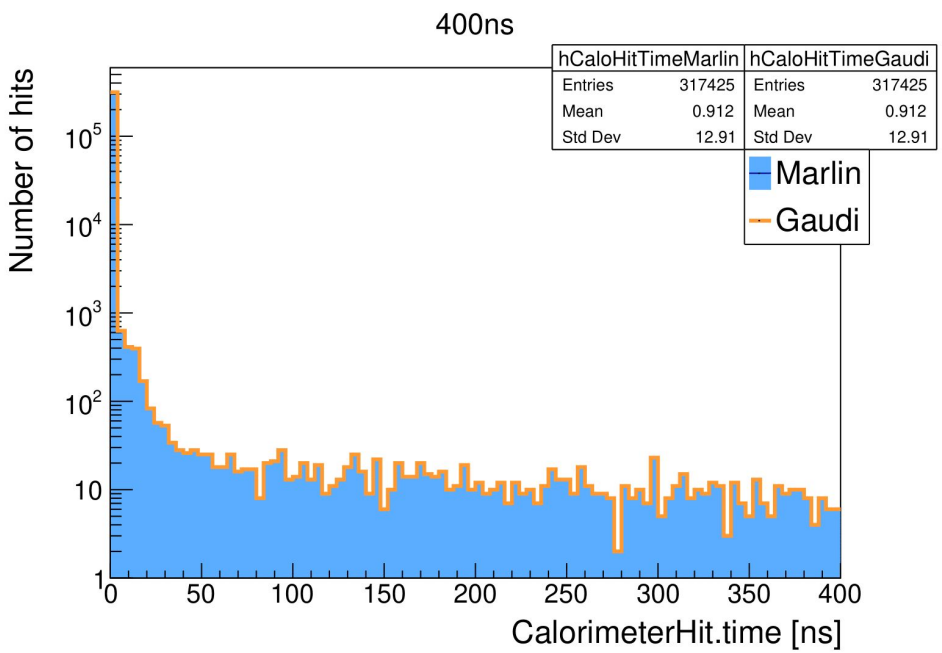
→ compare CalorimeterHit.energy for Marlin and Gaudi processors



DDCaloDigi

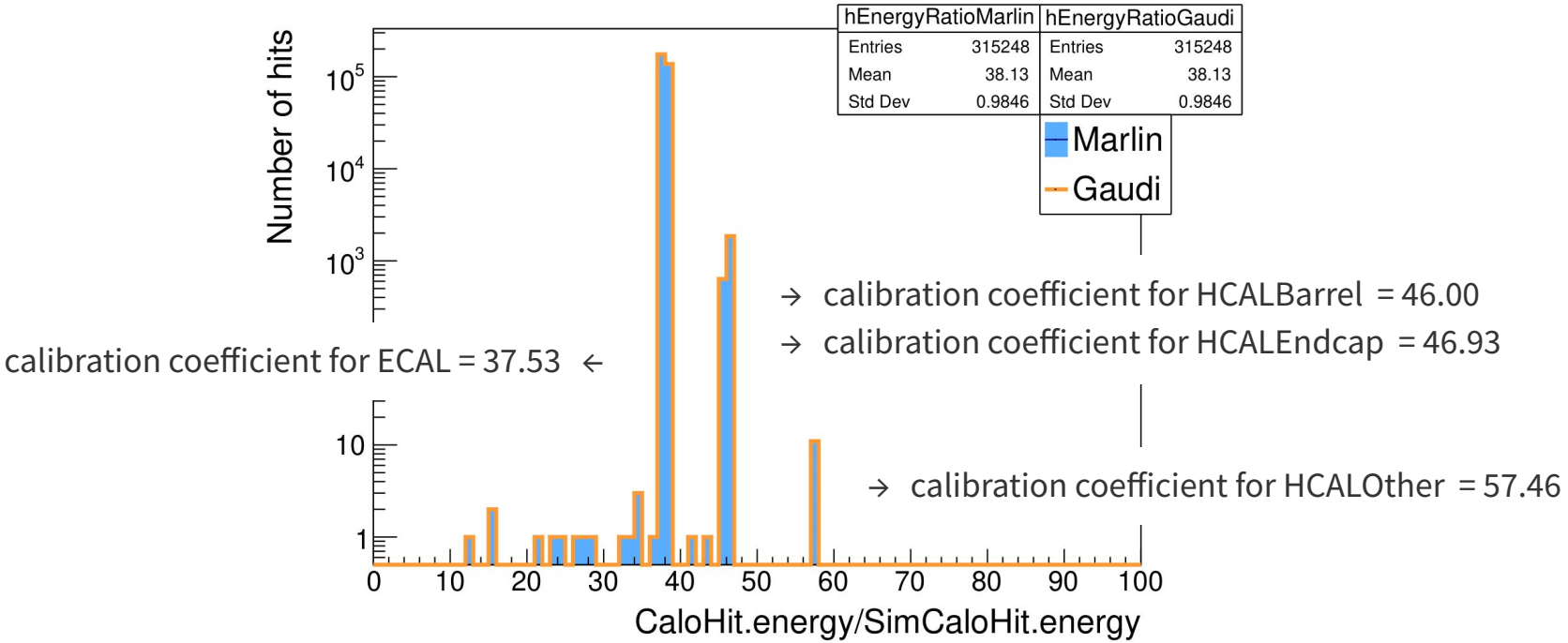
→ compare *CalorimeterHit.time* of Marlin and Gaudi processors





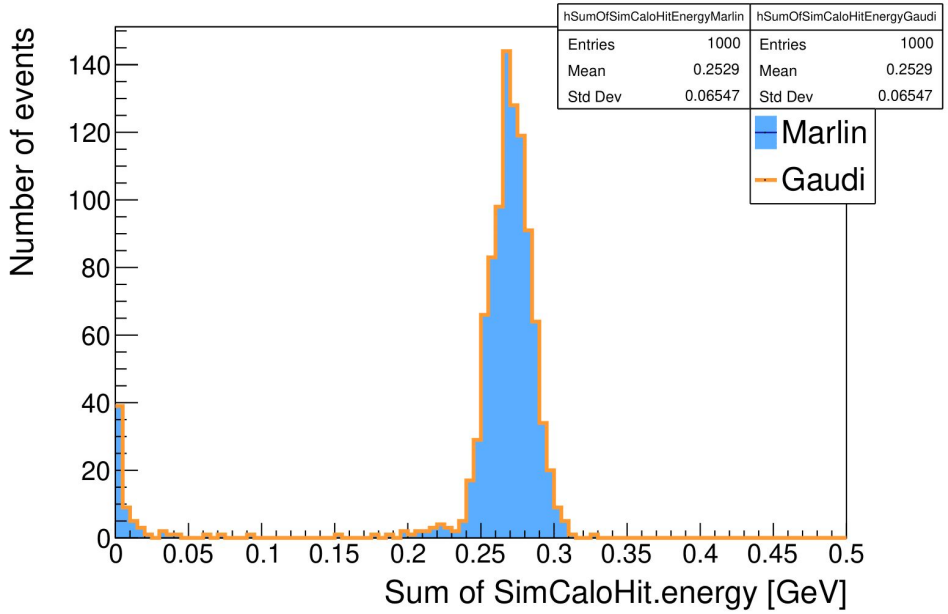
DDCaloDigi

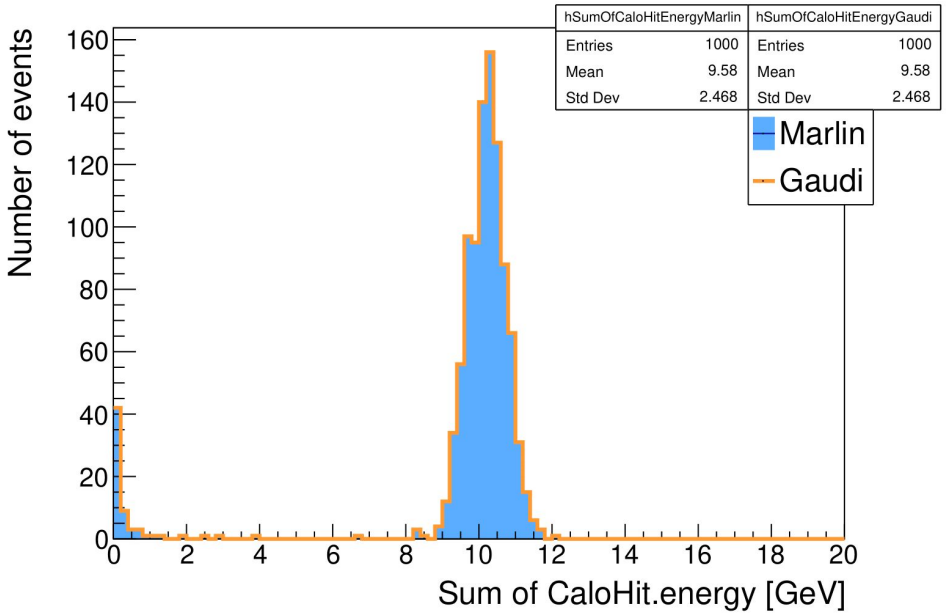
→ compare the ratio between *CalorimeterHit.energy* and *SimCalorimeterHit.energy* for every reconstructed hit (use `edm4hep::CaloHitSimCaloHitLinkCollection`)



DDCaloDigi

→ compare sum of *SimCalorimeterHit.energy* per event and sum of *CalorimeterHit.energy* per event



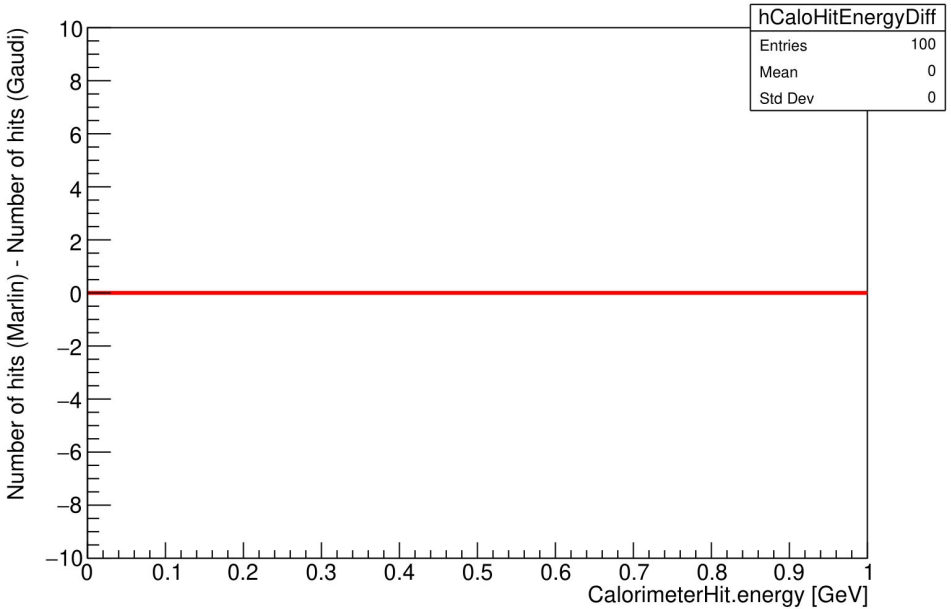


→ sum of *CalorimeterHit.energy* per event does match the simulated particle energy (10 GeV)

Conclusion and outlook

Output is the same from both processors. Porting is successful!

- Porting has been finished and validation has been done for the **DDSimpleMuonDigi** and **DDCaloDigi** reconstruction algorithms
- The ported algorithms has been added to the k4GaudiPandora repository
- TODO: finalise the PR for DDCaloDigi in k4GaudiPandora
- Future work: port and validate of *DDPandoraPFA* algorithm to Gaudi



Literature and useful links:

- Key4hep: Turnkey Software for Future Colliders - https://github.com/key4hep
- Key4hep documentation - https://key4hep.github.io/key4hep-doc/
- The Key4hep software stack: Beyond Future Higgs factories, 2023, https://arxiv.org/pdf/2312.08151
- Key4hep: Progress Report on Integrations, 2023, https://arxiv.org/abs/2312.08152
- k4GaudiPandora repository - https://github.com/key4hep/k4GaudiPandora
- DDMarlinPandora repository - https://github.com/iLCSoft/DDMarlinPandora

Thank you!

Contacts: katerina.kostova@cern.ch, kpkostova1@gmail.com