LCG Releases and Key4hep: Epic Stacks

André Sailer

CERN-EP-SFT

ePIC Software and Computing Meeting
April 26, 2024
CERN

Table of Contents

LCG Releases in a Nutshell

Providing compilers and consistent software stacks: $\approx$ 800 packages

- For a large set of architectures, operating systems, compilers
  - $\text{len}((x86, ARM) \wedge (\text{EL7}, 8, 9, \text{mac12}, \ldots) \wedge (\text{gcc11}, \ldots, \text{clang16}, \ldots) \wedge (\text{opt}, \text{dbg}) \wedge (\text{dev3}, \ldots)) \approx 50$
- ROOT, Geant4, MC Generators, ML packages, ...
- Every night (except Sundays) to `/cvmfs/sft[-nightlies].cern.ch`
- LCG releases and experiment specific stacks ("layers") are provided on CVMFS and as RPMs
- **Librarian and Integrators Meeting ("LIM")** every 2 weeks to discuss and decide on the content of nightlies and release dates
- Information/Documentation: **https://spi.web.cern.ch/**, **https://lcginfo.cern.ch**, **https://lcgdocs.web.cern.ch**

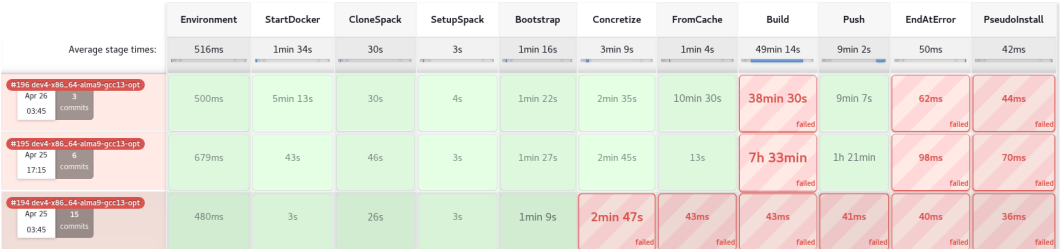
Building the LCG Software Stacks

- Builds are configured using the in-house **LCGCMake**
- For each package a build recipe describes the necessary steps and dependencies
  - Check if package with specific version+dependencies was already build, identified by hash
    - Attempts download of binary tarball from our binary repository
  - Otherwise build the package
    - Download the sources from our mirror (`URL`), using `cmake FILE` command
    - `Configure; make; make install` or equivalent
    - Upload the binary tarball to our binary repository

```
LCGPackage_Add(
  libgeotiff
  URL ${GenURL}/libgeotiff-${libgeotiff_native_version}.tar.gz
  IF <VERSION> VERSION_GREATER_EQUAL 1.6.0 THEN
    CMAKE_ARGS -DCMAKE_BUILD_TYPE=${CMAKE_BUILD_TYPE}
               -DCMAKE_INSTALL_PREFIX=<INSTALL_DIR>
               -DCMAKE_CXX_STANDARD=${CMAKE_CXX_STANDARD}
               -DBUILD_SHARED_LIBS=ON
               -DWITH_TIFF=ON
               -DTIFF_DIR=${tiff_home}
               -DPROJ_DIR=${proj_home}
  ELSE
    CONFIGURE_COMMAND ./configure --prefix=<INSTALL_DIR>
                                  --with-proj=${proj_home}
                                  --with-libtiff=${tiff_home}
    BUILD_COMMAND ${MAKE}
  ENDIF
  BUILD_IN_SOURCE 1
  DEPENDS proj tiff
  REVISION 2
)
```

Spack4LCG

- On going effort to use Spack for building LCG Stacks
- Replacing LCGCMake while in operation is tough
  - LCG Stacks are a moving target
  - Users might depend on peculiarities of existing build system, directory structure, or other assumptions
  - Updating, fixing, user support of LCG releases takes most of the work
  - Not enough time for developing workflows in Spack
  - Spack is also a moving target

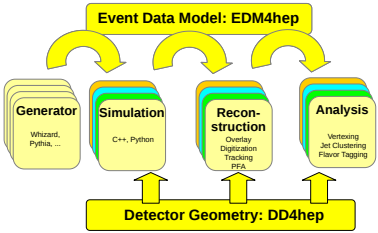


	Environment	StartDocker	CloneSpack	SetupSpack	Bootstrap	Concretize	FromCache	Build	Push	EndAtError	PseudoInstall
Average stage times:	516ms	1min 34s	30s	3s	1min 16s	3min 9s	1min 4s	49min 14s	9min 2s	50ms	42ms
#196 dev4-x86_64-alma9-gcc13-opt Apr 26 03:45 3 commits	500ms	5min 13s	30s	4s	1min 22s	2min 35s	10min 30s	38min 30s failed	9min 7s	62ms failed	44ms failed
#195 dev4-x86_64-alma9-gcc13-opt Apr 25 17:15 6 commits	679ms	43s	46s	3s	1min 27s	2min 45s	13s	7h 33min failed	1h 21min	98ms failed	70ms failed
#194 dev4-x86_64-alma9-gcc13-opt Apr 25 03:45 15 commits	480ms	3s	26s	3s	1min 9s	2min 47s failed	43ms failed	43ms failed	41ms failed	40ms failed	36ms failed

Key4hep

Key4hep

- Turnkey software for studies for experiments at future colliders
- Share components to reduce maintenance and development cost and allow everyone to benefit from its improvements
- Complete data processing framework, from generation to data analysis
- Major ingredients: Event Data Model (EDM4hep), Geometry Information (DD4hep), Processing Framework (Gaudi)



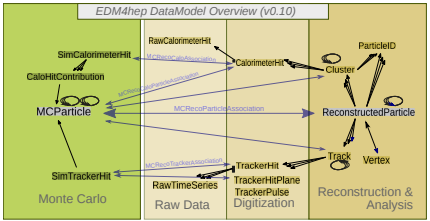
Building Key4hep with Spack

- Full Key4hep stack build with Spack
  - **https://github.com/key4hep/key4hep-spack**
  - Spack recipes for CEPCSW, FCCSW, iLCSoft
    - All communities can use the same stack
- Deployment to CVMFS (centos7, ubuntu22, alma9) with native compilers
  - Releases: `/cvmfs/sw.hsf.org/key4hep/`
  - Nightlies: `/cvmfs/sw-nightlies.hsf.org/key4hep/`
- Key4hep installation workflow stable
- Regularly updating spack branch

The Key4hep EDM: EDM4hep

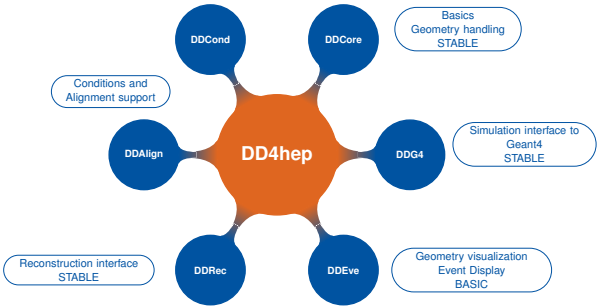
For a high degree of interoperability, EDM4hep provides a common event data model

- Using podio to manage the EDM (described by `yaml`) and easily change the persistency layer (ROOT, SIO, . . .)
- EDM4hep data model based on LCIO and FCC-edm
- **http://github.com/key4hep/edm4hep**
- Recent developments for podio or EDM4hep
  - podio: Interface types (e.g, tracker hits), basic schema evolution
  -
- Close to version 1 release (?)
  - Inverting ParticleID dependency, covariance matrix components



Geometry Information: DD4hep

- Complete Detector Description
  - Providing geometry, materials, visualization, readout, alignment, calibration...
- Single source of information → consistent description
  - Use in simulation, reconstruction, analysis
- Supports full experiment life cycle
  - Detector concept development, detector optimization, construction, operation
  - Facile transition from one stage to the next



- DD4hep already in use by CEPC, CMS, CLIC, EIC, FCC, ILC

Framework: Gaudi

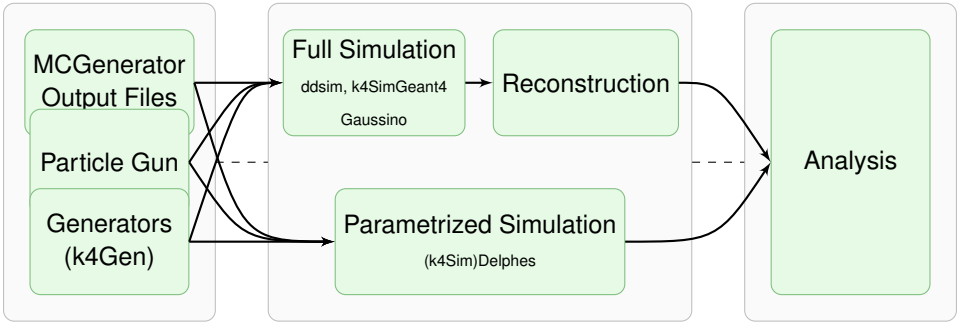
- Data processing frameworks are the skeleton on which HEP applications are built
- Gaudi was chosen as the framework, based on considerations for
  - portability to various computing resources, architectures and accelerators
  - support for task-oriented concurrency
  - adoption and developer community size; is used by LHCb, ATLAS
- Contribute developments were we see a need

k4FWCore

- Basic IO functionality: podio data service
- Interfaces for services

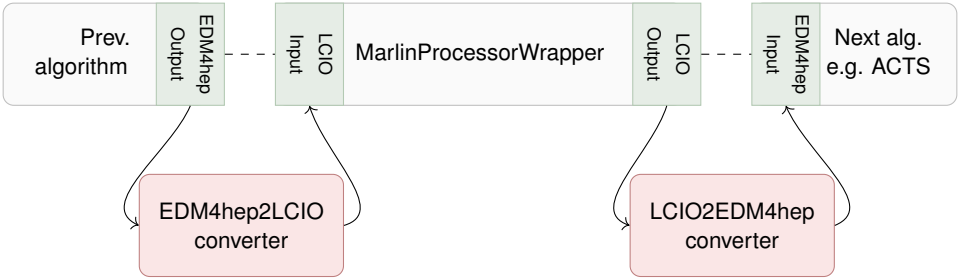
Simulation Integrations

- Key4hep allows to run fast parameterized simulation via Delphes, or Geant4 Simulation via DD4hep::ddsim (standalone) and k4SimGeant4 (Gaudi interface)
  - All solutions output data in EDM4hep format to be used in digitisation / reconstruction
  - Adoption of Gaussino planned as a replacement for k4SimGeant4, aligning-with/adopting functionality from DD4hep::DDG4



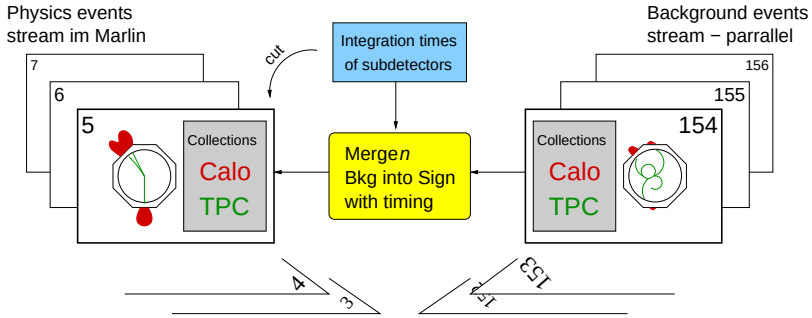
iLCSoft Integration

- k4MarlinWrapper to run iLCSoft Marlin Processors as Gaudi Algorithms
- In-memory conversion from different EDMs: LCIO <-> EDM4hep



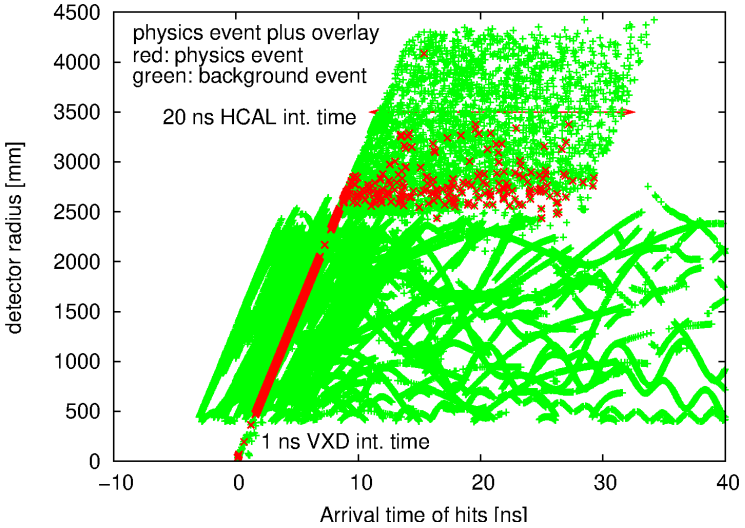
Background Overlay

- In iLCSoft there are multiple processors to overlay background events to physics events
  - Overlay: Overlay full background events (fixed number, Poisson distributed) to single physics event.
  - OverlayTimingGeneric [1]: Overlay background events with shifting T_0 to account for background from out-of-time events, apply timing window of detectors
- Gaudi and EDM4hep native implementation in the works



OverlayTiming(Generic)

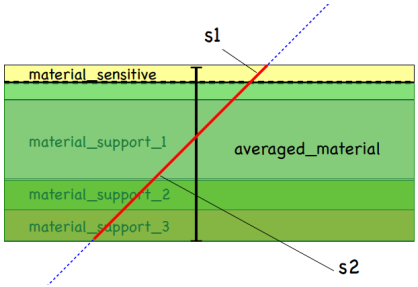
- Example output for overlaid event of CLIC_ILD_CDR (with a TPC main Tracker)
- Physics at $t = 0$ ns, some background events before, more after



Track Reconstruction

- iLCSoft tracking algorithms available through the *k4MarlinWrapper* approach
- Integration of the ACTS tracking toolkit as thin Gaudi Algorithm *ongoing*:

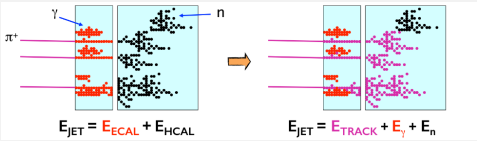
- Inject/add into ACTS surface information provided by `dd4hep::rec::Surface`
  - After the geometry instantiation, via DD4hep's plugin mechanism

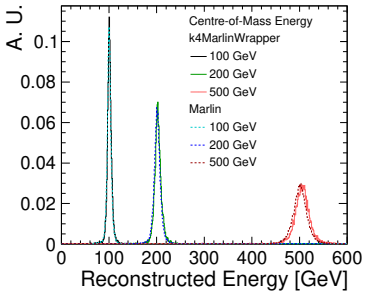


PandoraPFA: Particle Flow Clustering

PandoraPFA

Particle Flow clustering **toolkit** for high-granularity calorimeters. Currently available through the *k4MarlinWrapper* approach, validation wrt *Marlin* required.



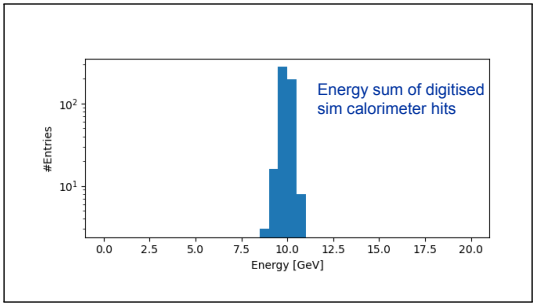


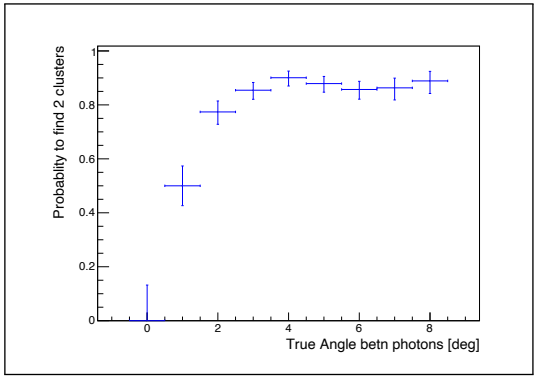
- Fairly consistent results for reconstruction comparing *k4MarlinWrapper* wrt *Marlin* including PandoraPFA
- Adaptation to new detector concepts, e.g. LAr-based, through DD4hep geometry drivers / plugins

Working on Gaudi based interface to PandoraPFA

PandoraPFA Reconstruction for LAr

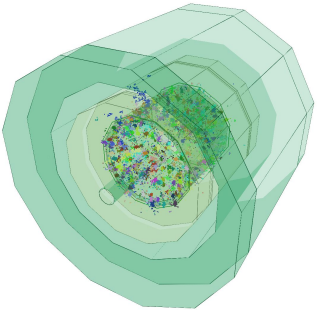
- Reconstructing photon clusters in the LAr ECal with PandoraPFA [2]
- Single particle(below), and two particle separation (right)
- Applying PandoraPFA in *completely* different calorimeter than original foreseen

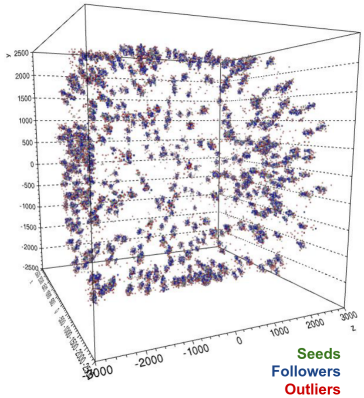




Clustering with Clue

- Clue (Clustering Energy) is a GPU friendly clustering algorithm developed for CMS HGCal [3]
- **k4CLUE** integration of Clue for the Key4hep stack able to reconstruct clusters in different detectors, more flexible approach digesting DD4hep information to come





Reconstruction of 500 Photons in CLD ECal

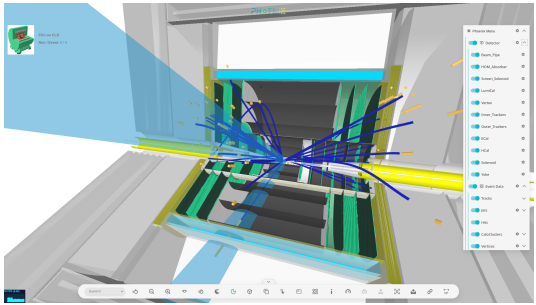
Analysis with RDataFrame

- EDM4hep data stored in ROOT Tree / RNTuple lends ideal candidate for analysis with RDataFrame
- Collection of tools in **FCCAnalyses**
- Example parts of the Higgs-Factory “Standard Candle” Higgs-Recoil analysis

```
theDataFrame
# define an alias for electron index collection
.Alias("Electron0", "Electron#0.index")
# define the electron collection
.Define("electrons",  "ReconstructedParticle::get(Electron0, ReconstructedParticles)")
#select electrons on pT
.Define("selected_electrons", "ReconstructedParticle::sel_pt(10.)(electrons)")
# ...
.Define("zed_leptonic_recoil_m","ReconstructedParticle::get_mass(zed_leptonic_recoil)")
# create branch with leptonic charge
.Define("zed_leptonic_charge","ReconstructedParticle::get_charge(zed_leptonic)")
# Filter at least one candidate
.Filter("zed_leptonic_recoil_m.size()>0")
```

Visualisation with Phoenix

- **Phoenix** an experiment independent web-based event display for HEP, used by ATLAS, LHCb, . . .
- Adapted for use in Key4hep
  - Detector DD4hep-based geometries converted with **JSROOT**
    - Some pruning and configuration still needed to make detectors look nice
  - Event Data exported as JSON
- Example: **https://fccsw.web.cern.ch/fccsw/phoenix/**



Conclusions

- Stacks::SPI team builds large software stacks for many platforms using LCGCMake: LCGReleases
- Stacks::Key4hep develops software for detector studies
- Key4hep stack has large overlaps with EIC software: spack, DD4hep, EDM4hep, ACTS
  - Contributions from EIC team to Key4hep packages gratefully received!

Thanks and Acknowledgements

Thanks to my colleagues
for contributing material to these slides!

This work benefited from support by the CERN Strategic R&D Programme on Technologies for Future Experiments **(CERN-OPEN-2018-006)**.

This project has received funding from the European Union's Horizon 2020 Research and Innovation programme under Grant Agreement no. 101004761.

Thank you for your attention!

References

[1] P. Schade and A. Lucaci-Timoce. *Description of the signal and background event mixing as implemented in the Marlin processor OverlayTiming*. CERN LCD-Note-2011-006. 2011.

[2] J.S. Marshall and M.A. Thomson. "The Pandora Software Development Kit for Pattern Recognition". In: *Eur.Phys.J.* C75.9 (2015), p. 439. DOI: 10.1140/epjc/s10052-015-3659-3.

[3] E. Brondolin, M. Rovere, and F. Pantaleo. "The k4Clue package: Empowering future collider experiments with the CLUE algorithm". In: *Nucl. Instrum. Meth.* A1061 (2024), p. 169100. DOI: 10.1016/j.nima.2024.169100.