EP-R&D

Programme on Technologies for Future Experiments

WP7 Software

Erica Brondolin, Placido Fernandez Declara, Paul Gessinger-Befurt, Javier Lopez Gomez, André Sailer, Dalila Salamani, Graeme Stewart for the WP7 team

CERN

EP R&D Days, 2022-06-20

Introduction - André Sailer

Challenges for Software and Computing

Challenges for HL-LHC

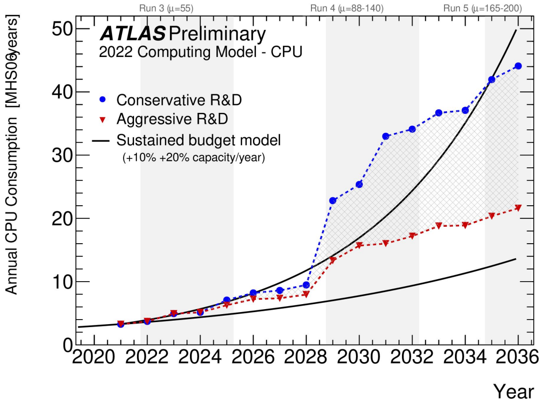
- Event rates
- Event complexity
- Precision physics

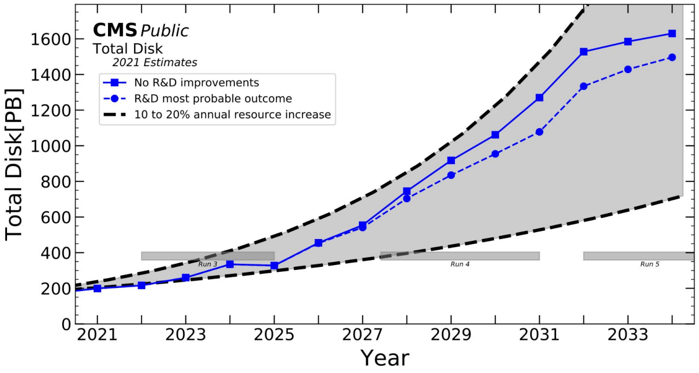
Challenges for Future Studies

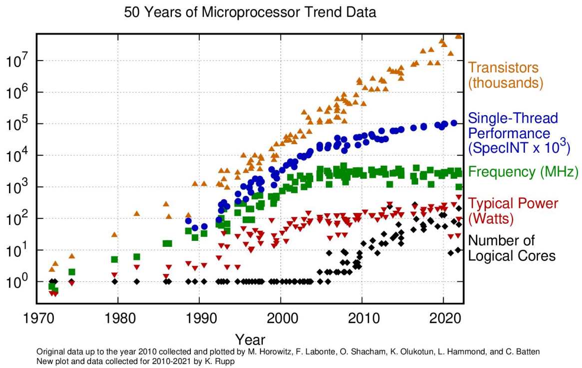
- Agile and sophisticated software
- High statistics

Plus... trends in

- Computing hardware
- Storage technology







ECFA Roadmap and Software

- The 2021 ECFA Detector Research and Development Roadmap
  - “Making software re-usable beyond a specific experiment or project [...]”
  - “The trends to exploit additional information [...] will require continuous refinement of the simulation tools [...] The same is true for sophisticated pattern recognition algorithms”
- Software remains a critical part of the design, preparation and exploitation of future detectors
- HSF Community Whitepaper: A Roadmap for HEP Software and Computing in the 2020s
  - “Establishing a programme of investment in software for the HEP community, with a view to ensuring effective and sustainable software for the coming decades, will be essential to allow us to reap the physics benefits of the multi-exabyte data to come.”

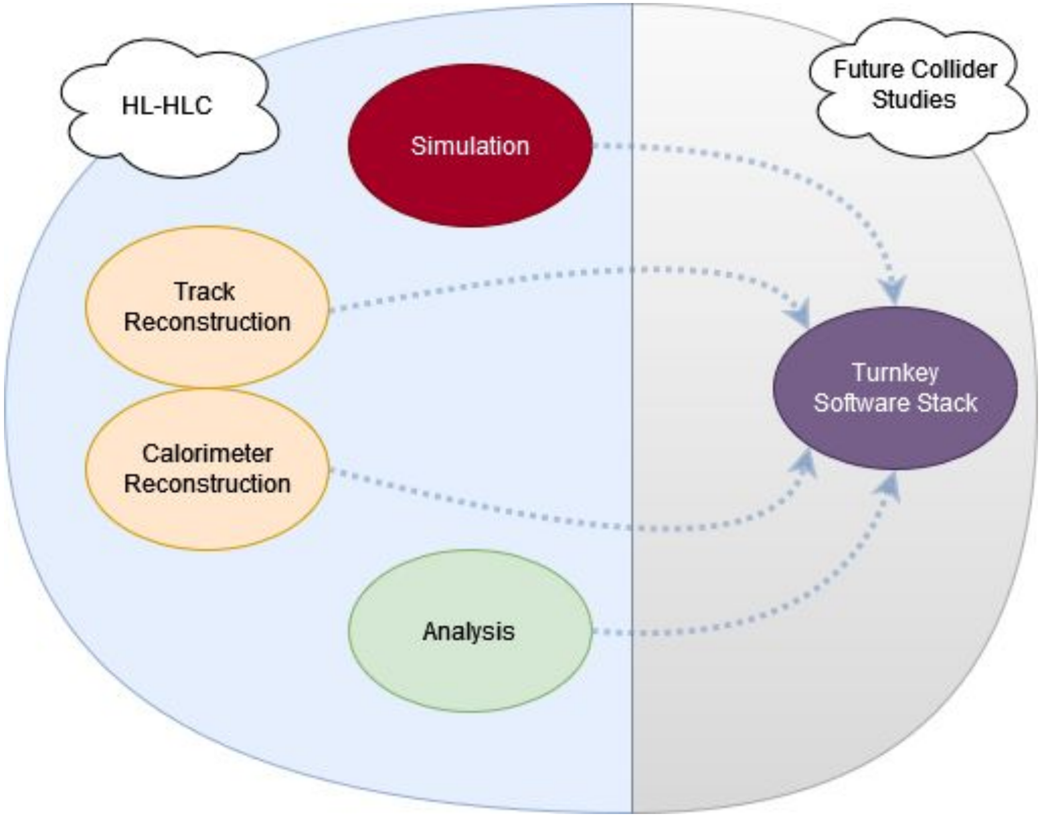
EP R&D WP7 Tasks

Address the challenges for HL-LHC of more and more complex events

- Faster Simulation
- Tracking and Calorimeter Reconstruction
- Analysis

Supporting Future Collider Studies with best of breed components

- Turnkey Software Stack
  - Testbed for other developments



Efficient Analysis - Javier Lopez Gomez

Introduction

- LHC runs 1 and 2 have seen great advances in analysis productivity; yet cannot handle an increase of an order of magnitude in recorded events at far greater complexity that upcoming particle colliders will bring.

- Analysis at this scale is crucial for the success of HL-LHC, and in general, CERN's scientific programme.

- Therefore, we address these challenges by:
    - (a) Increasing the data reading rate
    - (b) Forging programming models that boost scientists' productivity, e.g., by enabling transparent distributed execution of the same code in a distributed analysis facility.

Analysis R&D Implementation Path

The EP R&D analysis task is anchored in ROOT R&D activities:

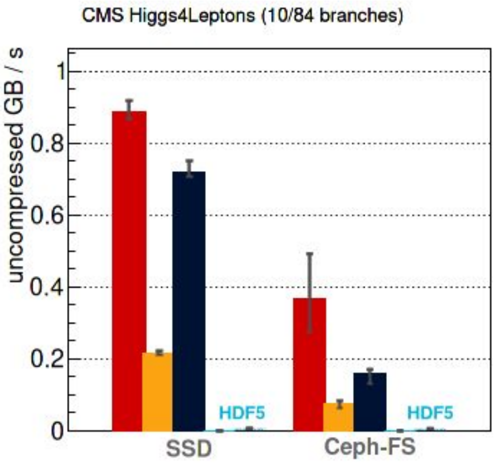
1. RNTuple: designated successor to the TTree I/O, redesigned for higher throughput, smaller data representation, and filesystem-less (e.g. object store based) data analysis.
    a. File-less storage systems are increasingly popular in HPC systems, where they can be used, e.g. to store data for a distributed analysis.

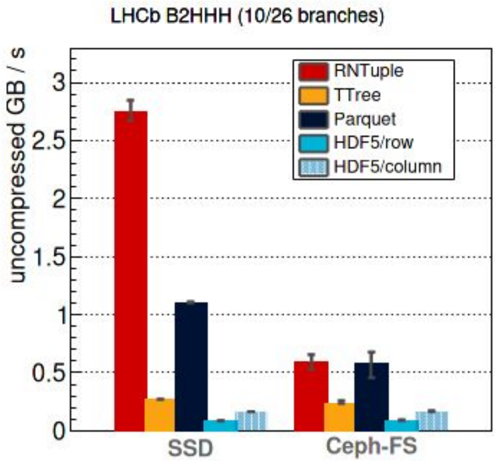
2. Development of a layer on top of ROOT's high-level analysis description system, RDataFrame, that makes distributed execution transparent for the end user.

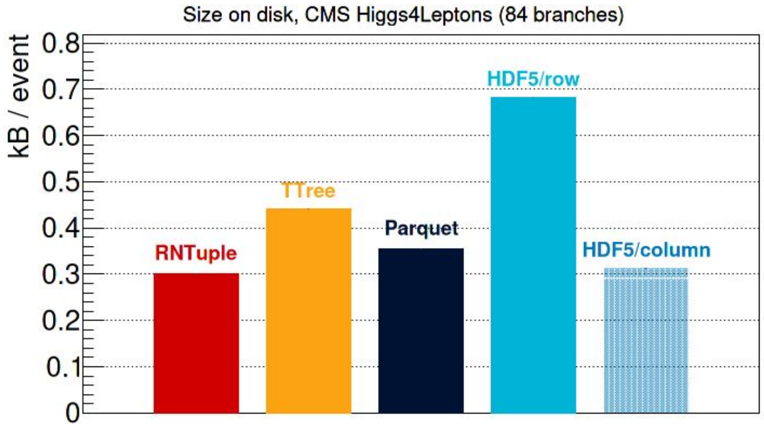
Months	Deliverables/Milestones
24	Study of storage access characteristics for data analysis with RNTuple *[Partially completed]*

Accomplishments I

- Performance engineering of the ROOT RNTuple I/O system
  - Targeting HL-LHC data rates and modern hardware platforms.
  - Provides significantly faster event throughput and smaller files than all the competitors (HEP + industry).
- Integration of RNTuple with HL-LHC frameworks
  - Collaboration with US IRIS-HEP: added support in the CMS core software framework (CMSSW).



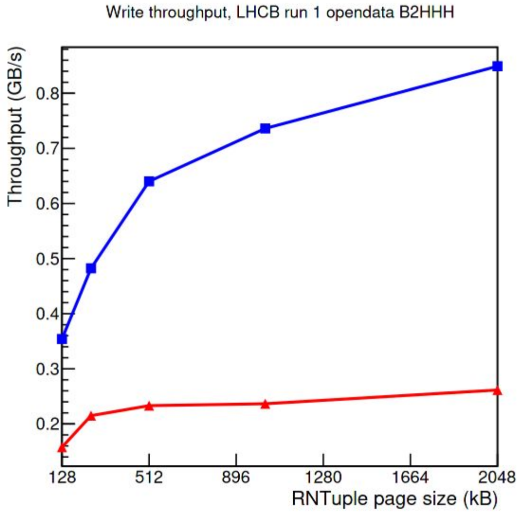


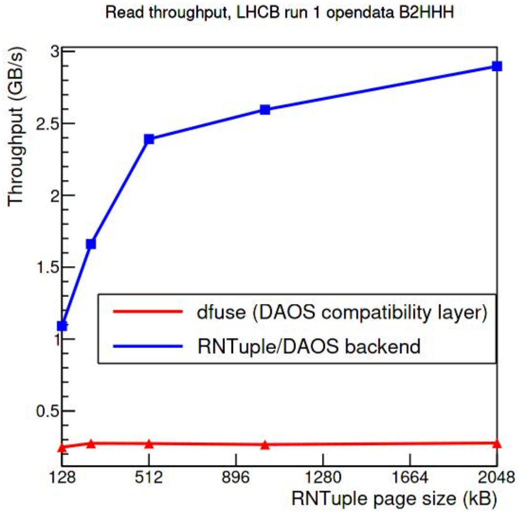


Accomplishments II

Months	Deliverables/Milestones
12	Prototype integration of ROOT's RNTuple IO with object stores
12	ROOT RNTuple IO subsystem able to read and write from/to object stores

- Object store support in RNTuple
  - Development of a viable backed for the Intel DAOS object store (EP R&D)
  - CERN openlab, Intel, and HPE provided support in form of hardware access.
  - Close-to-target performance of several GB/s end-to-end throughput for typical analysis tasks.
  - Acquired knowledge can be reused to support other object stores, e.g. Amazon S3.

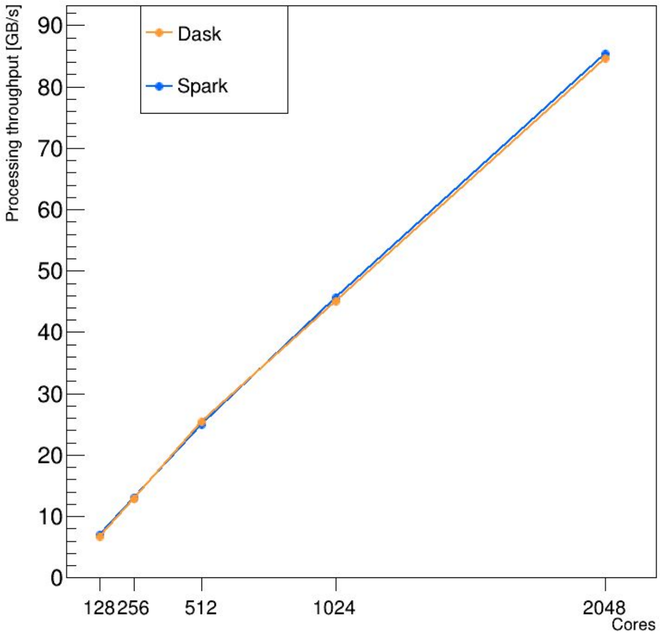




Accomplishments III

- A prototype of the ROOT distributed RDataFrame
  - Based on modern Big Data technologies (Spark, Dask).
  - More than 80 GB/s of real processing throughput on 2048 cores on a CERN HPC cluster.
  - Some experiments targeting distributed analysis where data is stored in DAOS, achieving up to 70% of the theoretical bandwidth of the cluster.
  - Also demonstrated that data can be transparently cached on an object store to accelerate further analysis on the same data.

Months	Deliverables/Milestones
36	Prototype cache engine for RdataFrame intermediate results
54 On-going	Implementation of a High-Throughput RdataFrame backend for closely-connected distributed systems *[Prototype]*
60 On-going	Fully functional integration of RdataFrame and RNTuple based analyses in a distributed environment *[Prototype]*



Processing throughput speedup of the Dimuon benchmark with distributed RDataFrame, comparing two different execution engines.

Next steps for 2022

The R&D on the RNTuple I/O demonstrates **significant performance gains** in read speed and data size both compared to current HEP production software and industry competitors.

Ongoing R&D on

- Write speed optimization to object stores: important for fast transfer of data from central storage systems to, e.g., object store enabled HPC sites
- Use of low-latency object stores as a data cache for distributed analysis workflows
- Integration with experiment frameworks and data models, e.g. ATLAS PHYSLITE
- Schema evolution: backwards and forwards compatibility of written datasets

Beyond 2022: focus on the efficient exploitation of **Analysis Facilities**

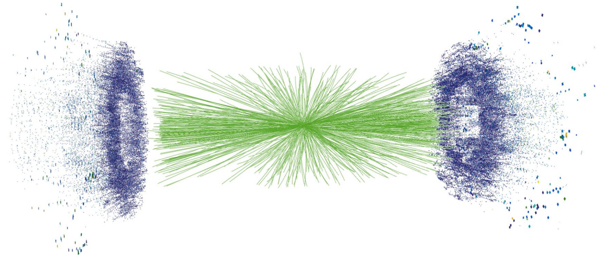
Reconstruction: Calorimetry - Erica Brondolin

Introduction

Months	Deliverables/Milestones
12	Review of existing clustering and reconstruction algorithms for concurrent execution, identification of potential and limitations of currently deployed algorithms and data structures. Development of mathematical framework for 6D track reconstruction.

- Particle shower reconstruction in **high-granularity calorimeter** is a very interesting and crucial task in high-density environments
  - Typical situation at HL-LHC ➔ Many showers tend to **overlap**
  - Standard reconstruction algorithms using combinatorics are expected to fail due to **memory/timing explosion**
  - **Fertile ground** for new techniques and algorithms: clustering, machine learning, graph theory, and modern computer architectures
    - Planned and designed, taking into account the information from the tracking system and timing detectors
    - Development can profit from experience with CMS Particle Flow techniques
- New **flexible framework** developed in CMS HGCAL which can be reused in other (future) experiments using high-granularity calorimeters

Tracks & Forward Calorimeter RecHits for single Electron event at 200 PU

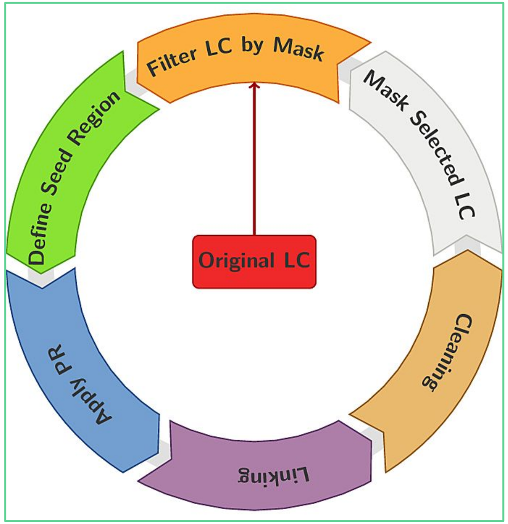


Months	Deliverables/Milestones
24	Extension with timing detectors including full 6D track reconstruction and including timing at clustering.

Reconstruction in CMS HGCAL

- TICL (The Iterative Clustering) is a **modular framework** integrated and under development in CMSSW
- Main **purpose**: processing calo rechits (x, y, z, t, E) and returning particle properties and probabilities
- In a nutshell: grouping 2D layer clusters (CLUE) into 3D clusters (Tracksters) **iteratively**
- Important features:
  - No prior knowledge of CMSSW needed to contribute
  - Modules are designed such that new algorithms or techniques (e.g. Machine Learning) can be plugged and swapped on top easily
  - Algorithms are designed with heterogeneous architectures / portability in mind
  - Mostly geometry independent

TICL iteration scheme

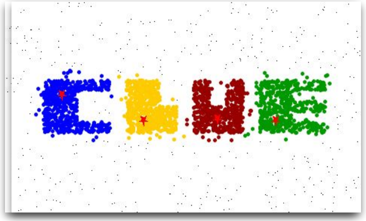


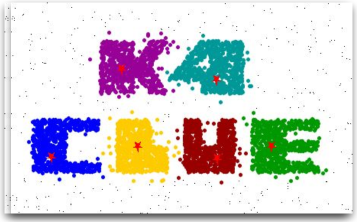
Documentation

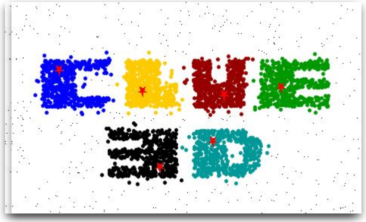
doi:10.3389/fdata.2020.591315

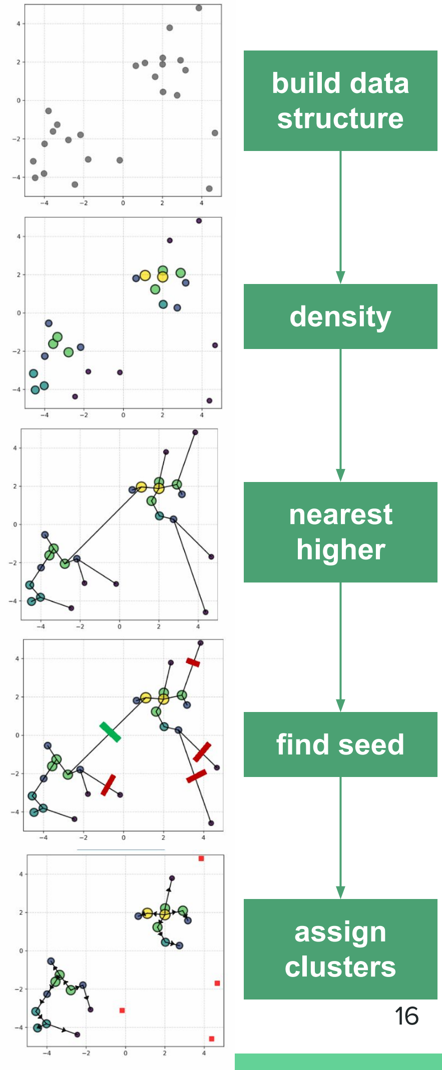
CLUE: CLUstering of Energy

- CLUE is a **fast and innovative** density-based clustering algo
- It groups energy deposits (hits) left by a particle traversing the active sensors of the calorimeter in clusters with a **well-defined seed** hit using the energy density - rather than individual cell energy.
  - Outliers, i.e. hits which do not belong to any clusters, are also identified.
- GPU-friendly, i.e. suitable for the upcoming era of heterogeneous computing in HEP



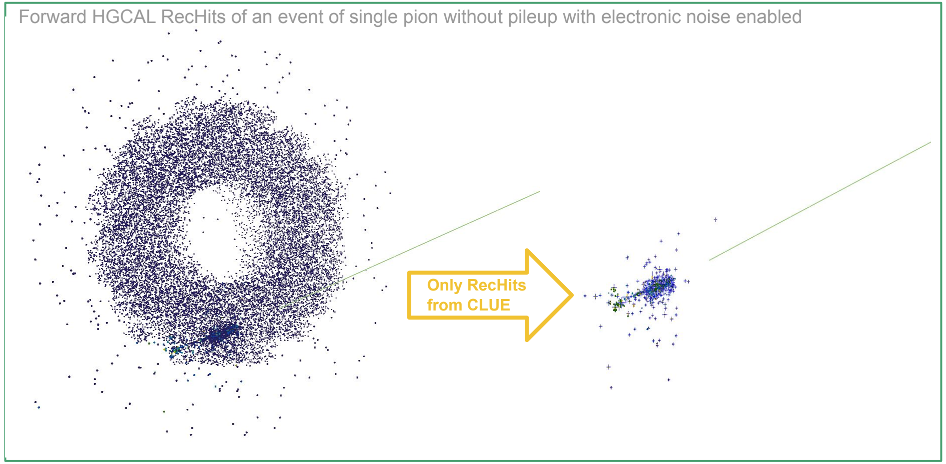


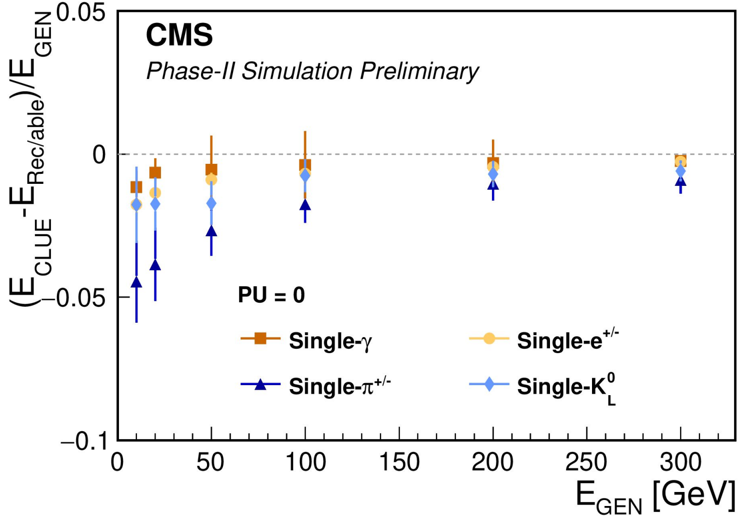




CLUE: CLUstering of Energy

- Events with about 200 pileup interactions at HL-LHC are estimated to produce an order of **10^5 RecHits** in the HGCAL detector
- CLUE clusters the RecHits in the same layer to produce **Layer Clusters** (LCs)
- CLUE (and full TICL) has been tested on electromagnetic showers and hadronic showers
  - Current approach: reconstruct the whole shower as a single trackster

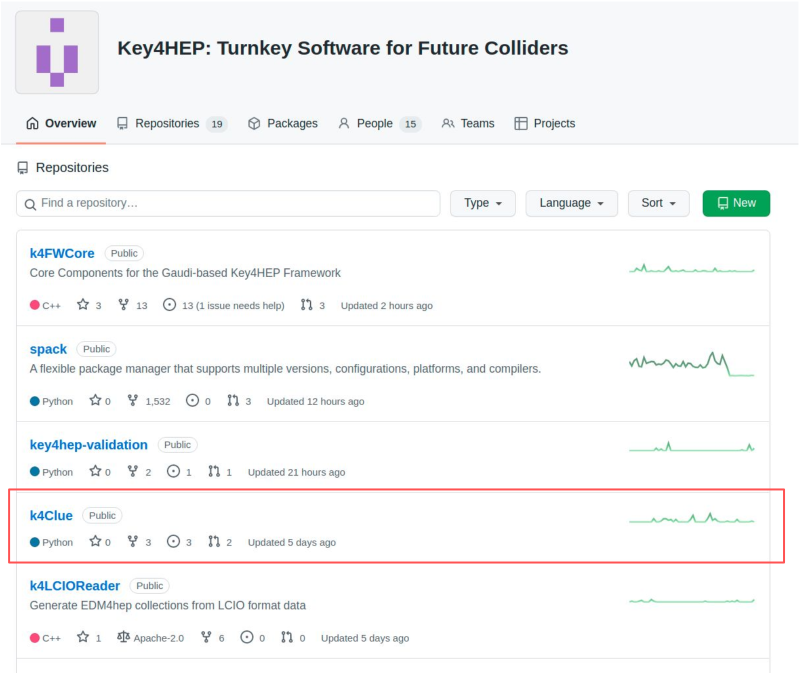




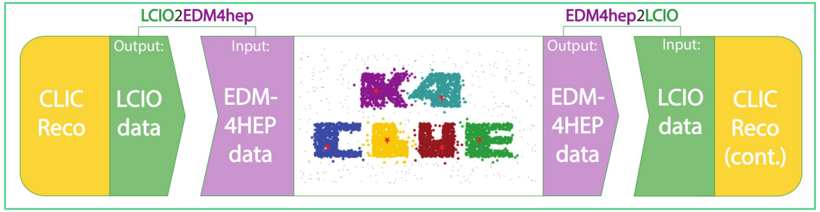
k4Clue

Months	Deliverables/Milestones
18	Prototype of standalone, concurrent, clustering algorithms library.
48	Investigation of GPU based clustering and prototype integration into high granular calorimetry for HL-LHC and the FCC-hh study.

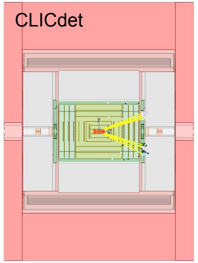
- CLUE was adapted to run in the Key4hep framework: k4Clue
  - It was integrated in the Gaudi software framework
  - Supports EDM4hep data format for inputs and outputs
  - Continuous Integration Tests on plain data and EDM4HEP data
  - Validation: New EDM4hep CLUECalorimeterHit class added with specific methods related to the CLUE algorithm
- In parallel, quite extensive work on extending standalone repo kalos/clue to be used as an external library



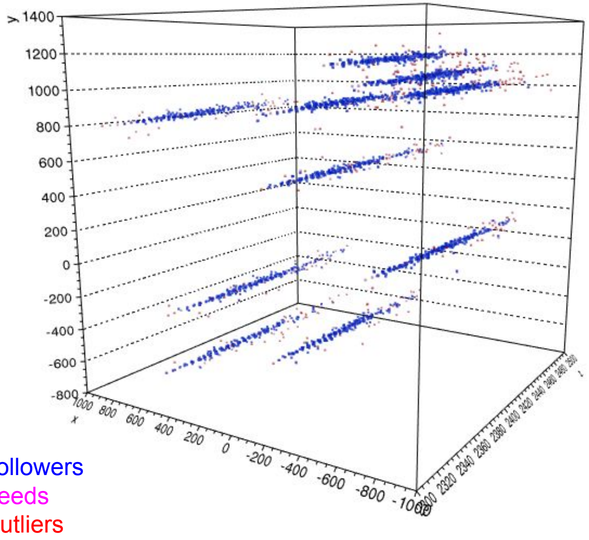
k4Clue

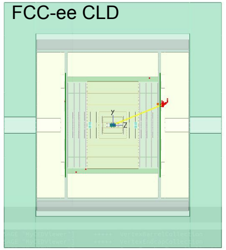


- k4CLUE was successfully run as part of the CLIC/FCC-ee CLD reconstruction chain thanks to the **excellent synergies** within the EP R&D group
- Validation on single electromagnetic showers **on-going**

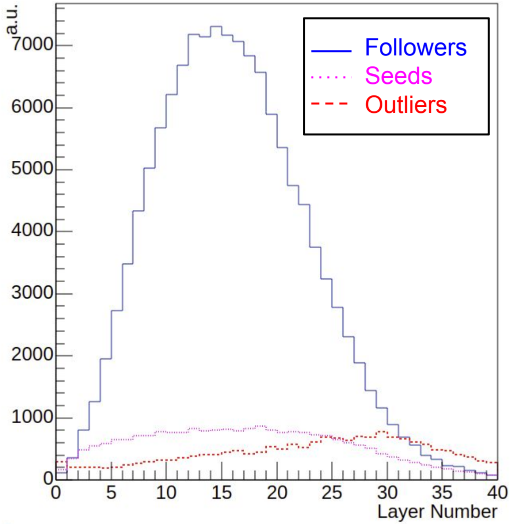


1 event with 10 single gammas at 10 GeV energy





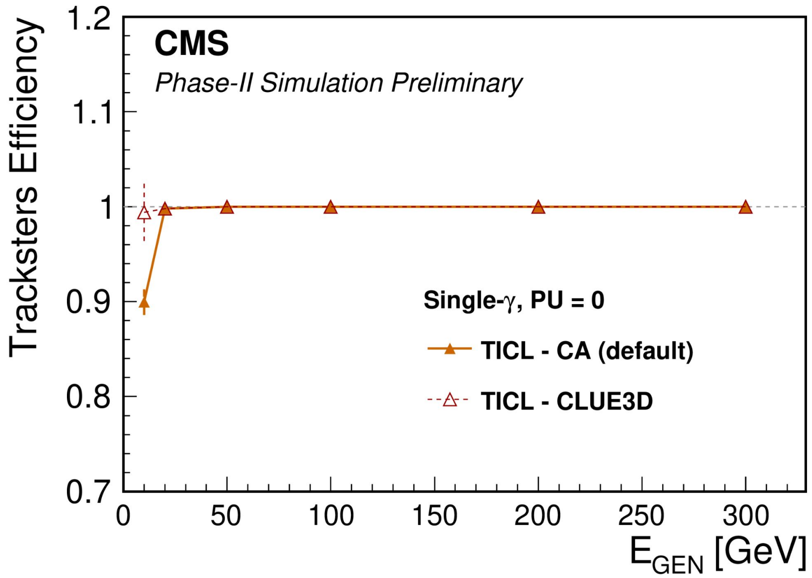
500 events of single gamma at 10 GeV energy



CLUE3D

- Novel **pattern recognition algorithm** in CMS HGCAL reco: CLUE3D
- It clusters LCs belonging to different layers in 3 dimensions using the same logic as CLUE
- Recently introduced **alternative projective coordinates:**

$$\left(\frac{r}{z}z_r, \frac{r}{z}z_r\phi\right)$$



- Discussion on-going on extend the CLUE library to incorporate CLUE3D

Months	Deliverables/Milestones
48	Investigation of GPU based clustering and prototype integration into high granular calorimetry for HL-LHC and the FCC study.
46	Characterization and performance comparison of different clustering algorithms, in different PU conditions and for several particle types (charged hadrons, electrons, photons, neutrals).

Next steps in 2022

- Performance assessment of K4 CLUE
    - Systematic studies to check physics performance and **parameters tuning**
    - Evaluate CLUE performance in **comparison** with the default PandoraClusters
    - Explore **visualization** techniques within the Key4hep framework
- Investigate the possibility of **implementing** CLUE 3D in the standalone repository of kalos/clue for fast and parallelizable pattern recognition
    - Its potential, optimal tuning and best usage are currently under study
- Explore **portability** of CLUE to different GPUs and CPUs using Alpaka and OneAPI software
    - Develop algorithms using portability frameworks and measure performance/scalability on different devices
    - Compare to native implementations

Reconstruction: Tracking - Paul Gessinger-Befurt

Track reconstruction challenge

Tracking is a **combinatorial problem** with non-linear scaling in compute cost

- **End of CPU frequency scaling*** affects scaling of reconstruction performance
- Software for LHC (and FCC/Future colliders) has origins in single-core era
- New **algorithmic approaches**, new **data model**, needed
  - Existing algorithm reviewed / adapted in light of new reality → **D12 & M9: ✅**
- Popular strategy: **parallel computing** (multi-core CPUs or accelerators e.g. GPUs)
  - Hand-written (re)implementation of algorithms to leverage parallel compute
  - Machine learning (and inspired solutions) can potentially solve this "automatically"

*: ~ single processors get faster automatically over time

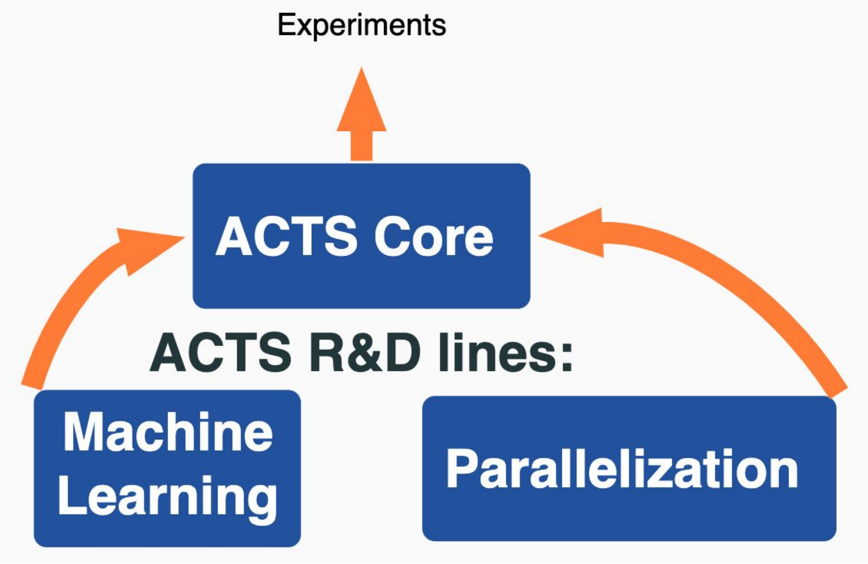
ACTS Core and R&D projects

A Common Tracking Software (ACTS) project established to collaborate on and evolve LHC state-of the art track reconstruction & build an R&D testbed.

Thread-safe algorithm design from the start!

Split into

- A **core project** targeting production (clients: ATLAS, sPHENIX, BGV, LDMX, …)
- Two **R&D lines**: machine learning & hardware acceleration (focus: GPU)

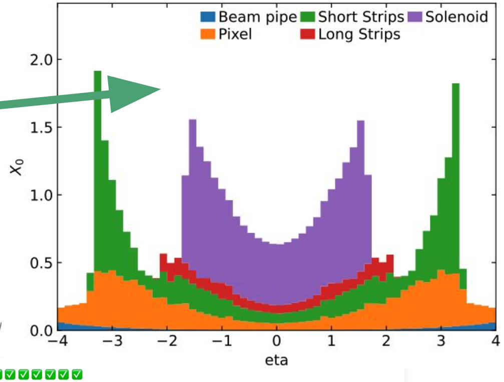


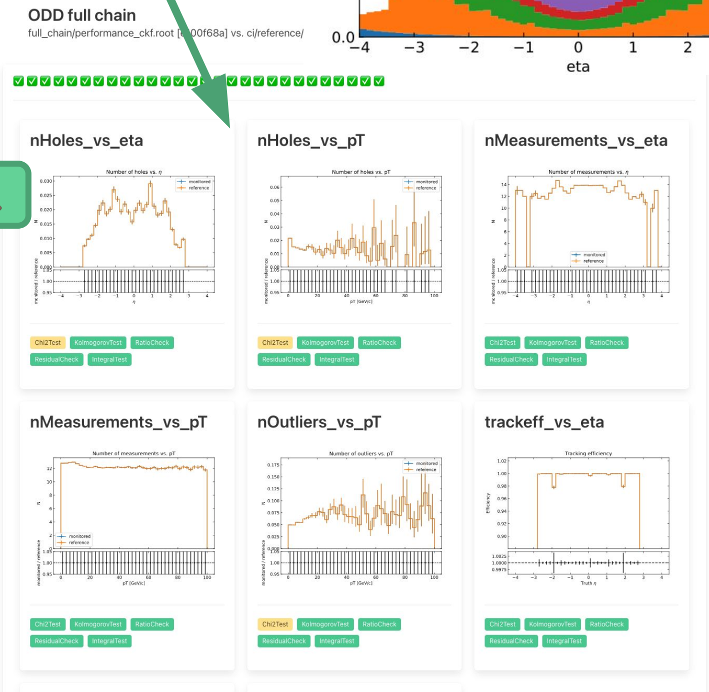
ACTS Core Library

- Contains all components for a basic* full tracking chain:
  - Silicon clusterization, pattern recognition, track finding, high precision track fitting
  - + Necessary lower level components: Detector geometry, particle propagation, magnetic field description
  - **Time information built-in & supported**
- Developments ongoing to complement / fill out feature set (specialized fitters, pattern recognition, etc.)
- Full-chain demonstrator developed on top of Open Data Detector, integration into Key4hep stack
- Comprehensive test infrastructure to monitor
  - Physics performance
  - ODD characteristics

D24 & M18: 🔨

Automated performance monitoring:





*: Typically strongly experiment-dependent components to be developed later

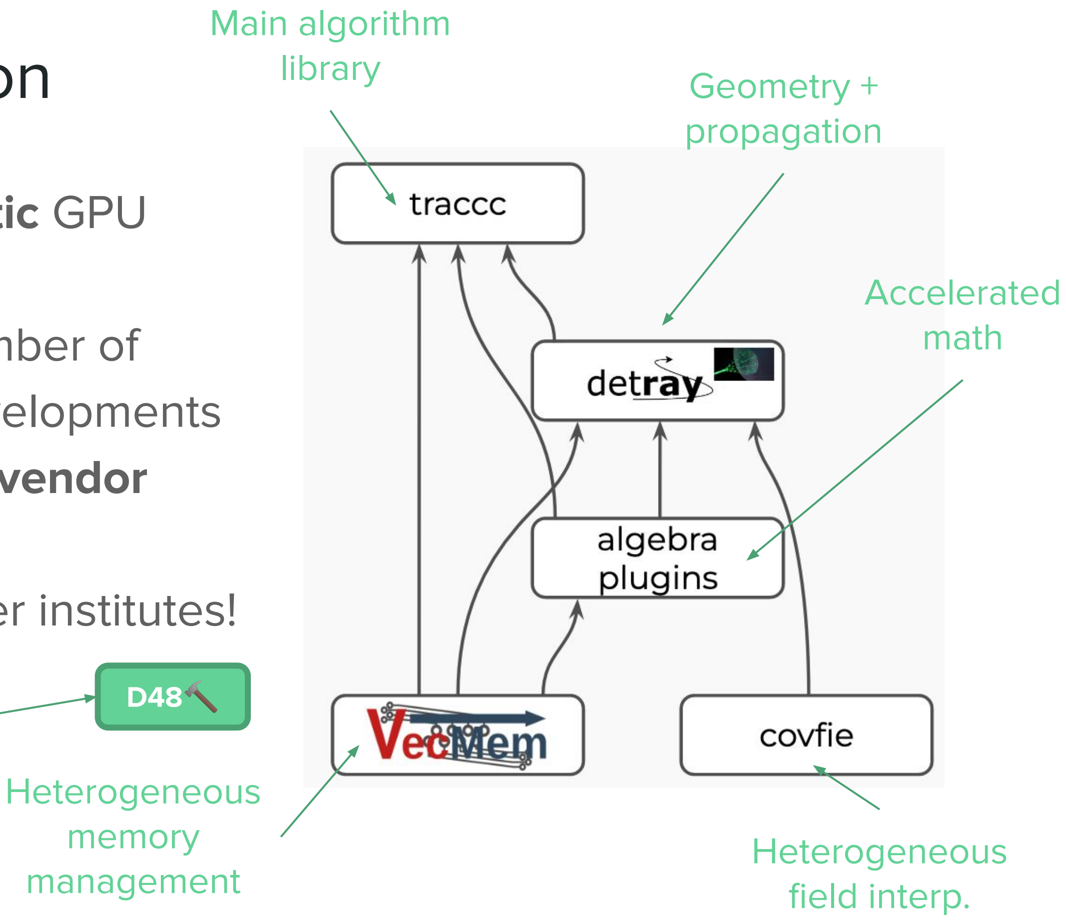
R&D Line 1 : Parallelization

- Goal: Demonstrator for **realistic** GPU tracking chain
- Started development of a number of **libraries** to support these developments
- Try to stay (GPU) **technology/vendor neutral**
- Strong collaboration with other institutes!

Category	Algorithms	CPU	CUDA	SYCL	std::par
Clusterization	CCL	✅	✅	✅	🟡
	Measurement creation	✅	✅	✅	⚪
	Spacepoint formation	✅	✅	✅	⚪
Track finding	Spacepoint binning	✅	✅	✅	⚪
	Seed finding	✅	✅	✅	⚪
	Track param estimation	✅	✅	✅	⚪
	Combinatorial KF	⚪	⚪	⚪	⚪
Track fitting	KF	🟡	🟡	⚪	⚪

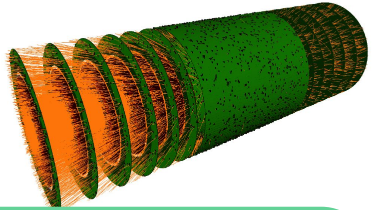
✅: exists 🟡: work started ⚪: work not yet started

D48🔨

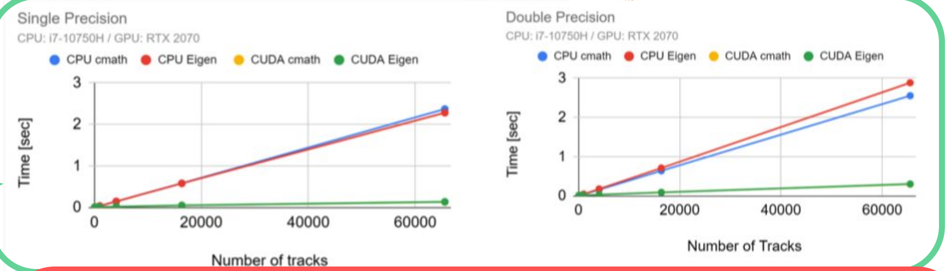


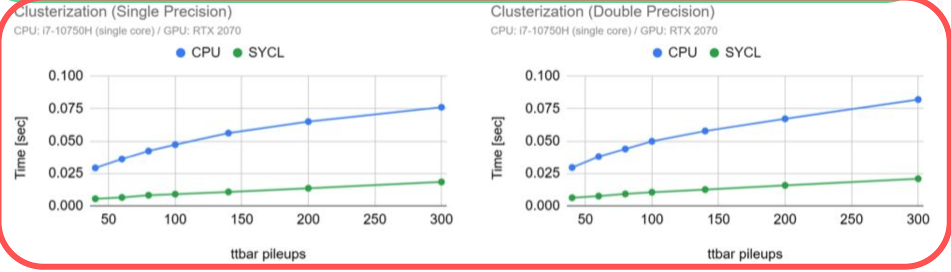
R&D Line 1 : Parallelization highlights

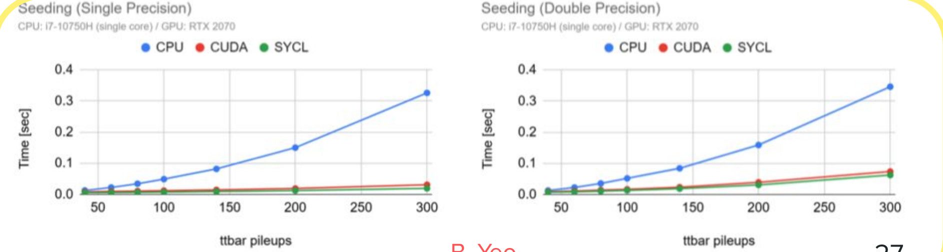
TrackML detector:



- First measurements from **numerical particle propagation** through test geometry on device:
  Scaling looks promising!
- **Clusterization** + spacepoint formation + **seeding** implemented
  - Performance again looks **promising**!
  - Observe benefits from running full chain on-device (less data transfer needed)
- Track finding + track fitting work ongoing

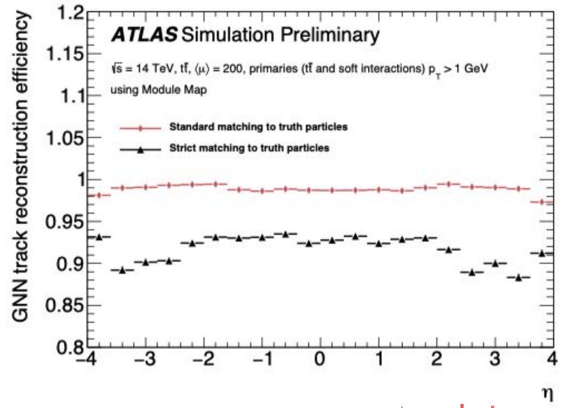




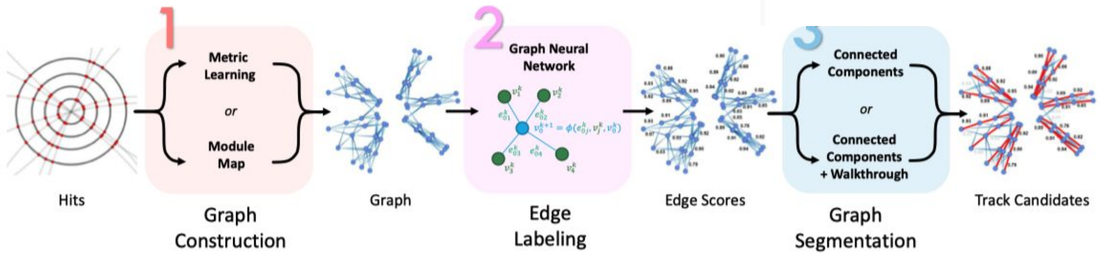


B. Yeo

R&D Line 2 : Machine Learning



- Collaboration with Exa.TrkX project:
  Graph Neural Network based track finding



plots

- **Learns** the geometry connections, labels likely track candidates
  - Physics + runtime performance becoming competitive!
  - Demonstrate ability to generalize to different detector (TrackML -> ATLAS ITk)
- Inference pipeline now **available in ACTS core** (work on training pipeline ongoing)

Prospects & plans

ACTS core library has evolved from early-stage prototype to approaching production-level quality

- Reimplements proven LHC algorithms in future-proof way + new algorithm development
- Understanding of performance improving continuously
- Open Data Detector serves as **performance reference platform**, also serves for community to develop algorithms against

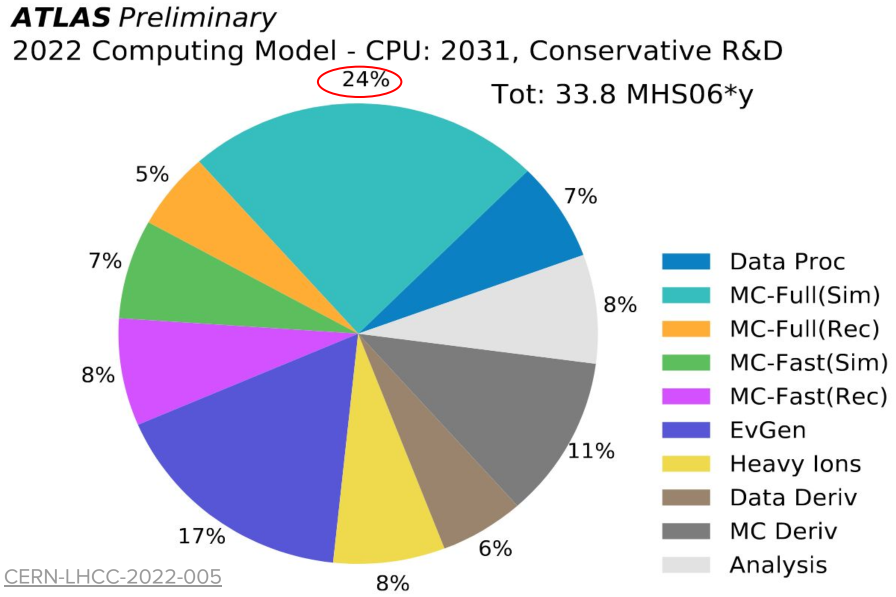
R&D lines explore alternative means of solving performance scaling

- Parallelization line aims to establish **realistic** tracking chain on GPUs.
  - Implemented up to particle propagation through geometry, track finding/fitting ongoing
  - Conclusively quantify performance characteristics
- Machine learning line aims to develop alternative algorithms
  - GNN based track finding being implemented as ready-to-use alternative in ACTS core
  - Move performance toward being competitive to conventional algorithms

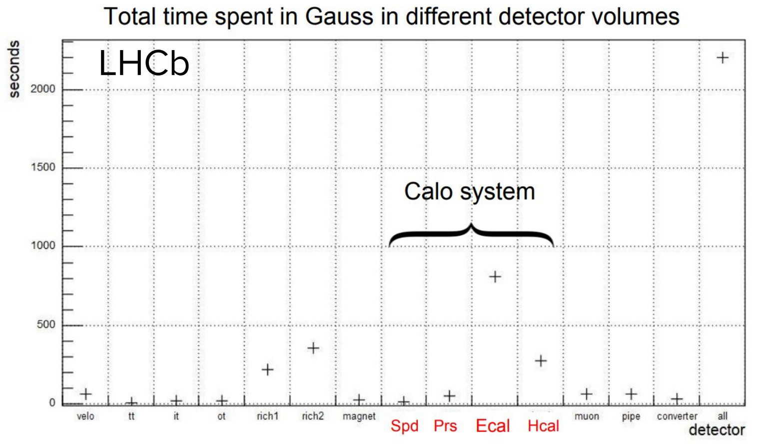
Faster Simulation - Dalila Salamani

Why simulation needs to be faster?

- Simulation takes substantial part of computing resources
- Calorimeters are the sub-detectors which usually are most time-consuming



CERN-LHCC-2022-005



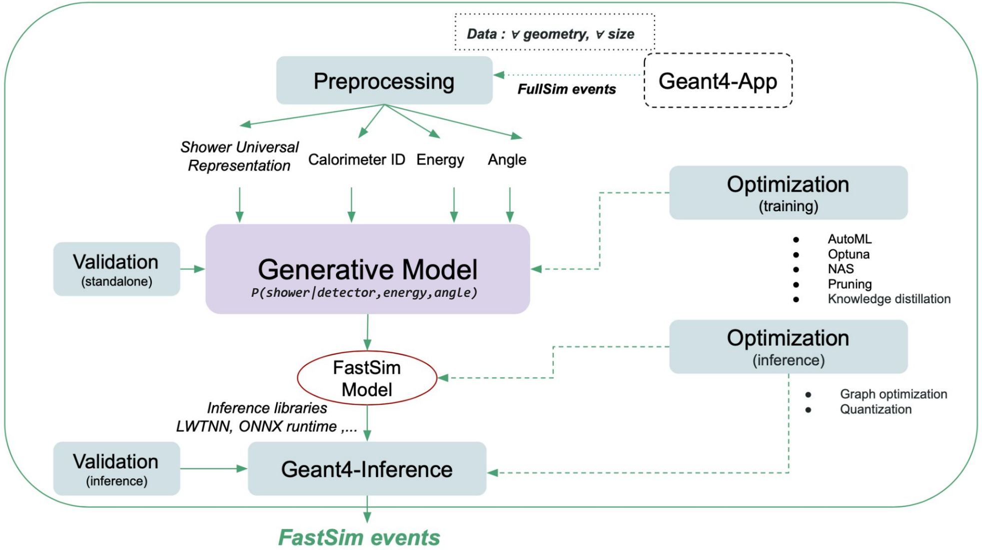
CPU time in calorimeter system: ~ **53%**

CHEP 2018, M. Rama

Towards general solutions for fast simulation

Focus on all aspects related to the fast-sim, systematic studies, reusable tools or methodology & systematic studies for HEP calorimeter fast-sim.

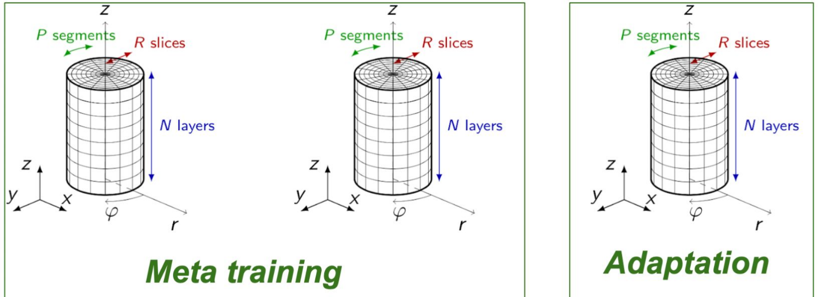
1. Data for the studies on different calorimeters (Geant4 example, Open Data Detector) and metrics to access quality (validation)
2. Meta-learning, Few-shot-learning
3. Hyperparameter tuning methods
4. Inference within C++ framework (Geant4)
5. Optimization - training, inference



Fast simulation ***documentation website*** *: https://g4fastsim.web.cern.ch/*

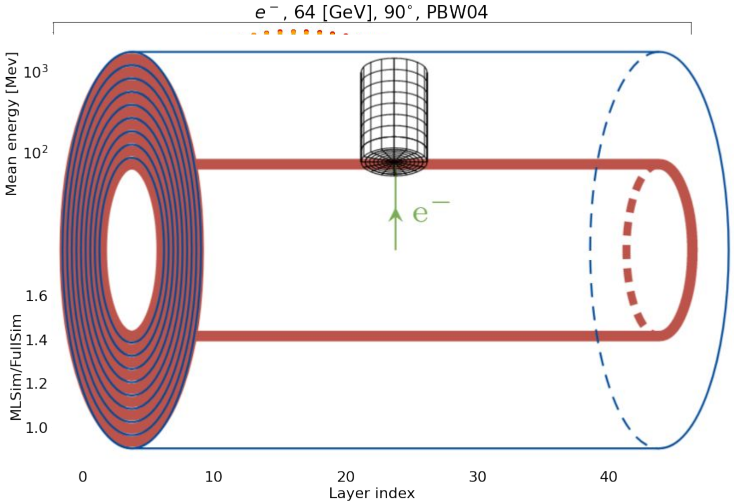
Meta-learning & fast adaptation

Months	Deliverables/Milestones
6	Review and evaluation of available ML generative models.
12	Investigation of DNN architectures for fast simulation.
18	Concrete detector fast simulation model



Material (s)	Tungsten and silicon	Lead and scintillator	Lead tungstate
Physical layer thickness	1.4 mm + 0.3 mm	4.4 mm + 1.2 mm	200.25 mm
RxPxN	18x50x45	18x50x45	18x50x45
Δr	2.3 mm	4 mm	4.9 mm
Δz	3.4 mm	5.6 mm	4.5 mm

Meta learning - Adaptation



Traditional training

- Meta training using geometries & adaptation on a new geometry
- **400 steps of adaptation : 20 s**

- Training on a single geometry with checkpoint saved every 100 epochs
- **400 steps of training : 20 min** (around 3h for 3900 steps)

Months	Deliverables/Milestones
30	Investigation of integration of fast simulation within full simulation toolkit.
36	Concrete detector fast simulation model

Integration with Geant4

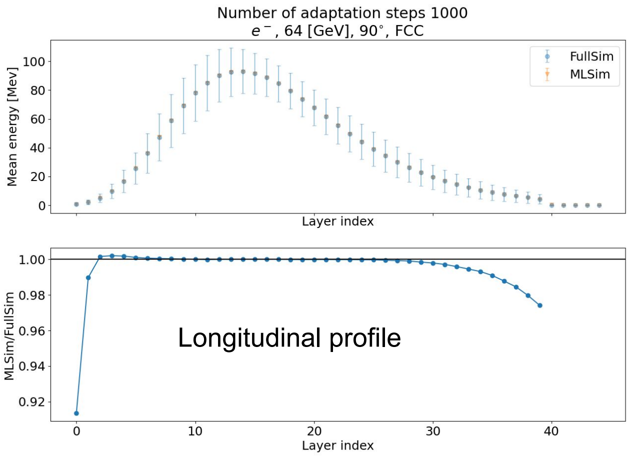
- Implementation of tools that facilitate use of fast simulation
- Development of Par04 example
  - Demonstrates how to incorporate inference libraries (ONNXRuntime, LWTNN)
  - The ML trained on two geometries, conditioned on the geometry, the energy and angle of the particle
  - Example can run full and fast simulation (if any of the inference libraries is available, e.g. via LCG)
  - Full simulation dataset is available on Zenodo and is part of the CaloChallenge

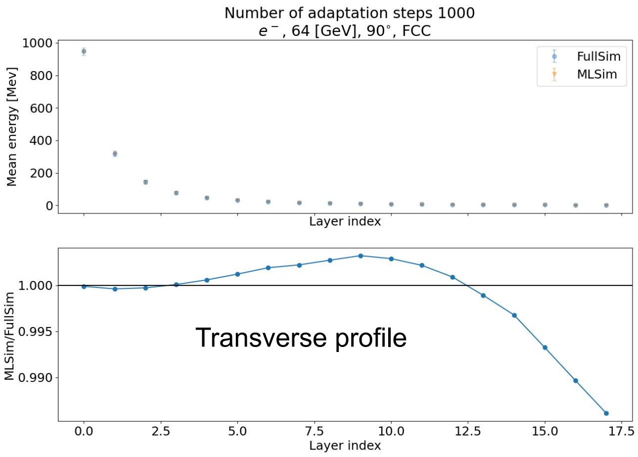
High Granularity Electromagnetic Calorimeter Shower Images

Fast Calorimeter Simulation Challenge 2022

Integration with Key4hep

- Integration of tools that facilitate use of fast simulation for future experiments
- FCCSW structure changes is based on Key4hep using k4SimGeant4
- Meta-learning model adaptation on CLIC-like detector for FCC-ee





The performance will improve once the model is pre-trained with more detectors

Plan of work 2022

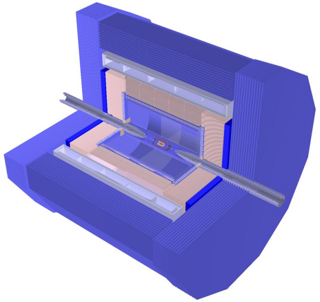
Months	Deliverables/Milestones
48 on-going	Implementation of different fast simulation models for various detectors.
48 on-going	Demonstration of generic fast-simulation tools for different detector classes
60 on-going	Optimisation tools for ML fast simulation

- Improvements of the ML model for highly granular calorimeters
- On-going work on validation for use on experiments geometries (FCC), with geometry-independent scoring
- Systematic studies on (meta-)training
- Studies of memory footprint optimization strategies (inference)
- Testing and development of optimized data pipelines (e.g. Kubeflow)
- Extension of Geant4 Par04 example (adding new inference libraries such as pyTorch C++API and TFLite, inference on GPUs)
- Follow-up of CaloChallenge with Open Data Detector calorimeter data
- Strong ties with AIDAinnova fast simulation task

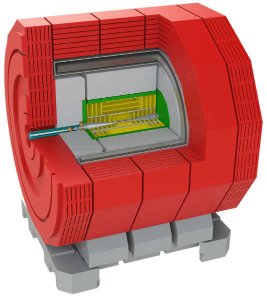
Turnkey Software Stack - Placido Fernandez Declara

Introduction

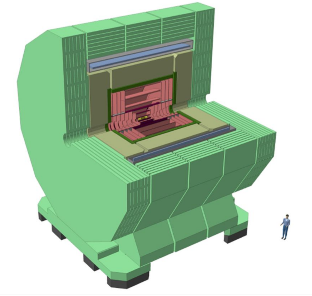
- Traditionally experiments at HEP have not done too well in sharing efforts towards having common tools and practices with software.
- Turnkey Software Stack goal is to provide a ready to use stack for FCC and CLIC communities, and other experiments: Key4hep.
- It is open to starting, planned and future experiments.



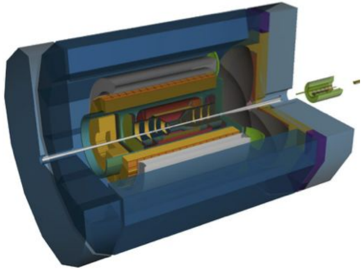
CEPC



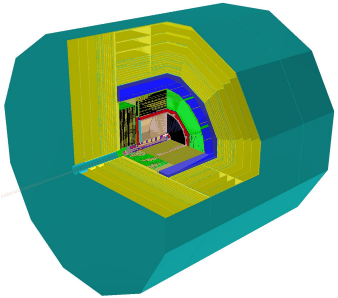
CLIC



FCC-ee



EIC

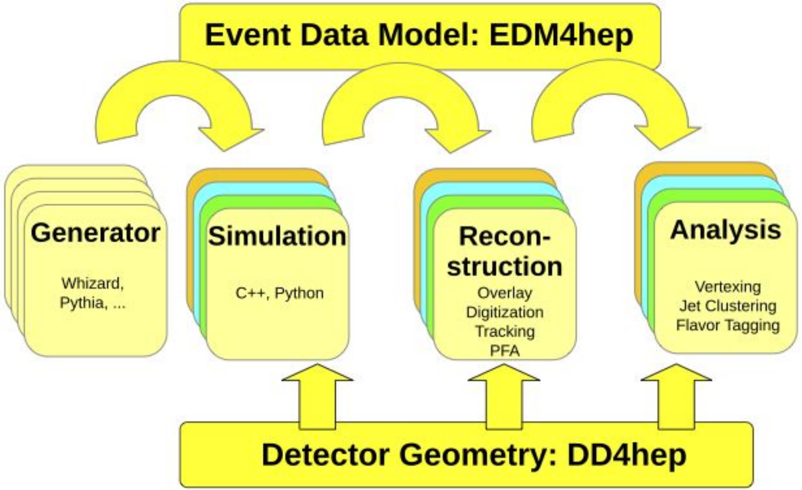


ILD

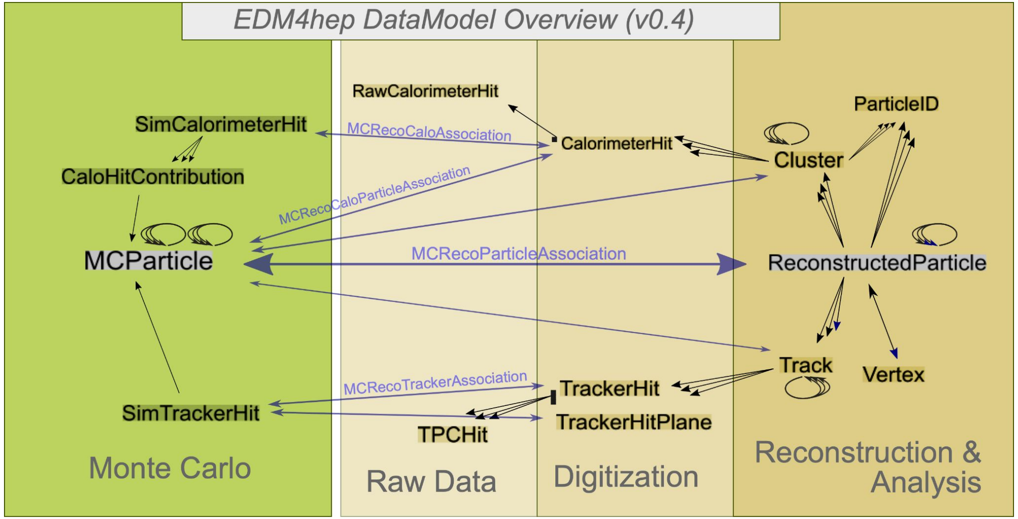
Overview of the solution

Single software stack that connects and extends packages to provide a complete data processing framework, comprising fast and full simulation, reconstruction and analysis.

- Data model: EDM4hep (podio)
- Common framework: Gaudi
- Geometry information: DD4hep
- Common simulation approach
- Streamlined process for software delivery:
  - Spack, CVMFS
- Common practices, tools, standards



Accomplishments I



- EDM4hep - podio
  - Common Event Data Model
  - schema evolution, converters, extension, metadata
- Common Gaudi framework
  - Used to schedule and steer simulation, reconstruction, analysis.
  - Already used by other institutes in Key4hep (CEPC, EIC)
  - Adapters (k4MarlinWrapper) provided to interface with CLIC's framework and algorithms
- Towards common simulation
  - Using k4SimGeant + k4SimDelphes
  - Moving towards using Gaussino taking the best of all

Accomplishments II

- Adaptors for CLIC to Gaudi framework
  - k4MarlinWrapper: EDM converters, input file converter, Marlin-Gaudi interface
- Validation of physics performance
  - Initial validation of CLIC reconstruction comparing original to Key4hep software stack results
  - WIP to add more validation tests and improve continuous integration
- Distributed computing
  - Updated Gaudi application integration in iLCDirac
  - Integrated FCC Virtual organization into iLCDirac
  - Developed transformation workflows for FCC in iLCDirac

month	Deliverable
18	Streamlined common stack for detector development, with adaptors for CLIC to Gaudi framework

month	Deliverable
48	Validation of physics performance in the new stack: fully functioning simulation and reconstruction system for some of the current CLIC and FCC detectors

Accomplishments III

- Turnkey software stack used and to be used by other experiments and communities
    - FCC/CLIC (Switzerland), ILC (Japan), CEPC (China), EIC (USA).

month	Deliverable
60	Turnkey software stack for the use by other experiments

- Integration with other WP7 tasks
    - CLUE: k4Clue
    - ACTS: k4ActsTracking
    - RNTuple (ROOT): initial backend for podio
    - OpenDataDetector: added to Key4hep stack

Next steps for 2022

- Moving towards running CLIC reconstruction without adaptors
  - Some algorithms will be fully ported and modernized
  - Others will rely on k4MarlinWrapper due to the simplicity of usage

month	Milestone
36	Run CLIC reconstruction using Gaudi without adaptors

- Running PandoraPFA for FCC-ee detector model also with LAr Calo (WP3)

month	Milestone
42	Run Pandora PFA in FCC-ee studies

Future Prospects and Plans - André Sailer

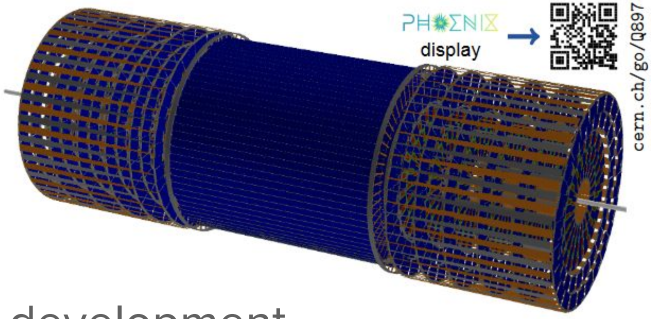
Future Directions for HEP Software R&D

- Directions identified by the HEP Software Foundation Community White Paper
- There is significant investment in software from the experiments, as well as other projects
  - EP R&D, AIDAinnova (EU), IRIS-HEP (US), SWIFT-HEP (UK)
- We work aligned with other projects, as well as with other CERN activities
  - EP-SFT, EP-ADP, EP-CMG
- Software work package is delivering crucial improvements for upgraded detectors
  - Great progress accomplishing milestones and deliverables we set out in 2020
  - This includes HL-LHC as software R&D to deployment cycles are shorter

Strengthening Support for Current Tasks

- Each of the current tasks is providing a significant contribution in its domain
  - We strongly support continuing these themes into a 2024-2028 funding cycle
- Combination of staff, fellow and student effort within a larger group is working well
  - Good integration into teams
  - R&D in all successful projects is a given, EP R&D provides a significant boost
- Additional funding for the projects would provide a specific ability to
  - Ensure that project knowledge between fellows is transferred most effectively
    - E.g., overlap between fellow positions
  - Provide extra support for PhD student projects
- Each software task has its specific aspects
  - Additionally, we generate skills which go beyond each single project
  - E.g., Key4hep integration, data model design

Expanding our Envelope



- We aim to develop additional competence for CERN as a world leading institute for HEP software development
- Clear areas where we could contribute more
  - Expertise in accelerators and heterogeneous computing
    - Specific needs of running HEP software on GPUs demand solutions for
      - Geometry representations
      - Data models
      - Workflow management
    - Would link up strongly with development of particle tracking on GPUs
  - Open Data Detector
    - Experiment neutral meeting point
    - Test bed for algorithmic development
    - Allows us to reach beyond traditional HEP community (TrackML, Calo Challenge) into data science

Summary

- The software work package of EP R&D has an impressive list of achievements
  - Highlighted in the 2021 R&D Report
- Success is measured in adoption of our projects for future initiatives
  - HL-LHC upgrades
  - FCC studies
  - Nuclear physics community, sPHENIX and EIC
- We are part of the roadmap towards future HEP facilities and experiments
- Continuing, and expanding where possible, this R&D is extremely important

Thank you for your support and thank you to everyone working in the R&D projects!