Strategic R&D Programme on Technologies for Future Experiments

CERN

Experimental Physics Department

WP7 Software

Graeme Stewart, Anna Zaborowska, Marco Rovere, Andi Salzburger, Jakob Blomer, Lorenzo Moneta, Benedikt Hegner for the WP7 team

CERN EP R&D Day

2023-02-20

Introduction

Trends in HEP Software and Computing

Software is ubiquitous in HEP

- Event generation, simulation, trigger, reconstruction, analysis

Challenges for future experiments

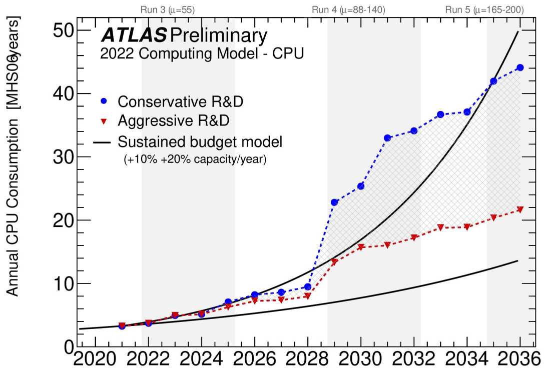
- Event rates
- Event complexity
- Precision physics

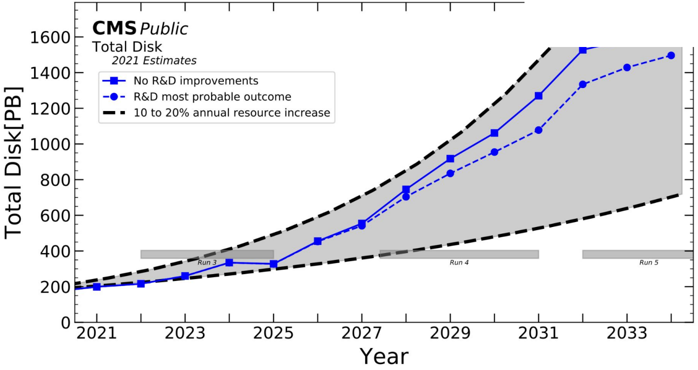
Challenges for Future Collider Studies

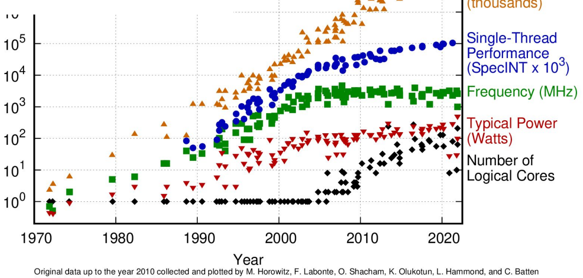
- Agile and sophisticated software
- Speed and accuracy of algorithms
- High statistics

Plus… difficult trends in

- Computing hardware
- Storage technology
- Energy costs, a.k.a., Green Computing



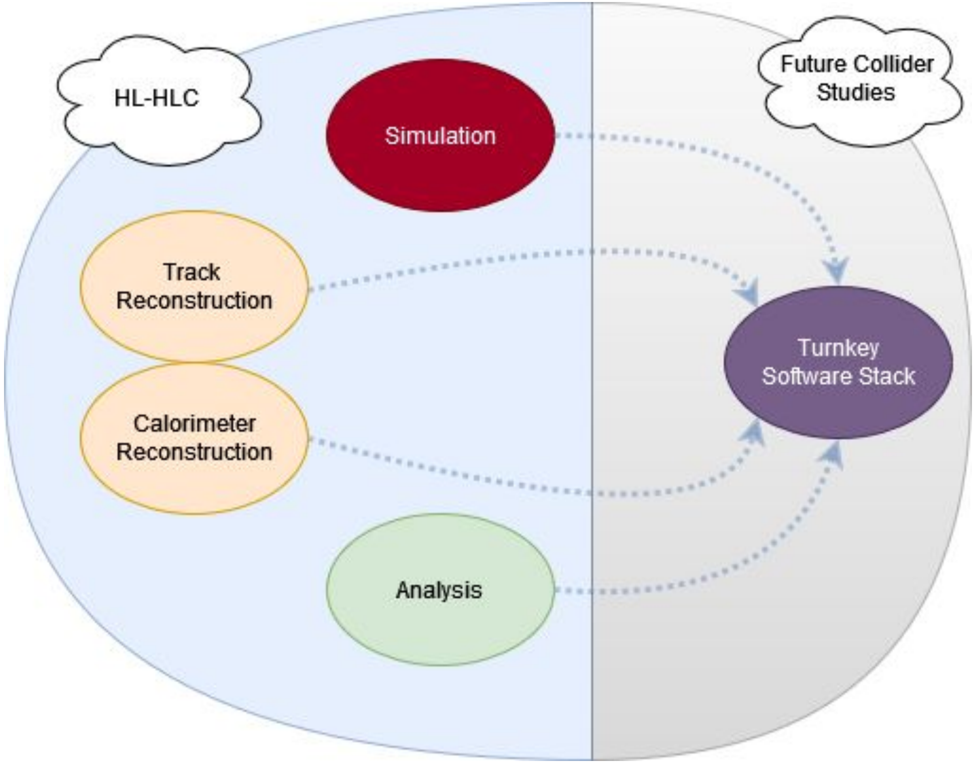




Original data up to the year 2010 collected and plotted by M. Horowitz, F. Labonte, O. Shacham, K. Olukotun, L. Hammond, and C. Batten
New plot and data collected for 2010-2021 by K. Rupp

EP R&D Phase I

- Address key components of future needs
  - Faster Simulation
    - *Including GPU particle tracking with AdePT**
  - Tracking and Calorimeter Reconstruction
  - Analysis
- Supporting Future Collider Studies with ‘best of breed’ components
  - Turnkey Software Stack - this is a testbed for other developments
  - *Modernisation of Gaudi framework for heterogeneous resources**



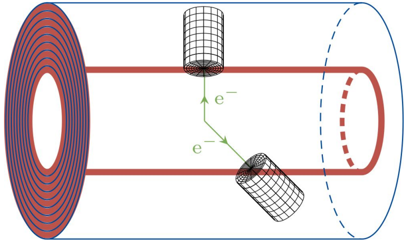
*New for 2023

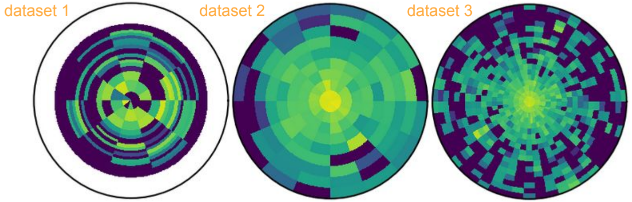
Towards Phase II

- Timescales
    - Software programme that still that applies to HL-LHC and the continual software evolution that will occur
    - Future upgrades for LHCb and ALICE (Run 5) enter more into scope
    - Continue to support more and more refined future detector studies
- Key motivations for software in the period 2024-2028 remain
    - We continue the primary thrust of many of the tasks
    - Some evolve significantly, e.g., in Turnkey moving to more specific framework R&D
    - Common themes develop and are an important unifying feature of the R&D proposal
        - Open Data Detector
            - Neutral testbed for algorithm development and open datasets
        - GPU Support and expertise
            - Building knowledge across tasks in how to use and support GPU code
            - HEP geometry, field maps, core mathematical operations, etc.
        - Hardware diversity - shared resources between tasks

Faster Simulation

Phase I: achievements





2 years of work within EP-RD:

- Implementation of necessary tools within Geant4 for Machine Learning (ML) fast sim (used in Gaussino, LHCb)
- Integration of inference libraries to demonstrate full ML fast sim cycle within Geant4 (as Geant4 example Par04, available in 11.0 release)
- Development of a Variational Auto-Encoder model for detector-readout independent simulation
- MetaHEP: generic ML fast shower model, able to retrain quickly to new detectors
  - tuning to detector is always necessary to obtain accuracy, but is much better than training from scratch
  - demonstration on simplistic calorimeters and one of the FCC-ee proposed calorimeters
- Publication of Par04 data on zenodo, co-organisation of first ML fast sim calorimeter challenge for shower model benchmarking and development
- On-going work with other WP7 tasks on Open Data Detector, a benchmark detector for algorithmic studies
- On-going work on a transformer-based generative model, with a goal of obtaining a more accurate fast shower simulation. Co-organised a dedicated workshop with CERN and IBM contributions.
- Cooperation with external groups within AIDAinnova, and CERN IT/openlab

Phase II: plans

Continuation of fast simulation developments:

- Demonstration of application of generic ML-based fast shower simulation (MetaHEP) to LHC and (other) FCC detectors
  - MetaHEP shows considerable speed-up of training: it is not done from scratch, but tuned to detectors
- Development of more accurate ML model for calorimeters
  - Current focus on transformer-based generative model, conclusions may be drawn before start of Phase II and those results will dictate the direction we take
- Encapsulation of ATLAS fast simulation code into experiment-independent package
  - Investigation of its use to future experiments

Phase II: plans

Addition of a new sub-task, targeting utilisation of GPUs for particle transport:

- Current status of the Accelerated demonstrator of electromagnetic Particle Transport (AdePT), a project that started > 2 years ago
  - G4HepEM is used to model EM physics interactions (also improves Geant4)
  - VecGeom is used to navigate particles in complex geometry (also improves Geant4)
  - Propagation of charged particles in a magnetic field
  - Simple generation of hits, which are transferred back from the GPU to the CPU
  - A standalone demonstrator and Geant4-integrated example
  - *Cooperation with SWIFT-HEP*

Phase II: plans

AdePT
Accelerated Particle Transport

Addition of a new sub-task, targeting utilisation of GPUs for particle transport:

- Validation and debugging of the current AdePT prototype
  - standalone and integrated into Geant4
- Addition of a non-constant magnetic field propagator with a realistic LHC experiment field map
  - validation and optimisation
- Completion of a new surface-based geometry model, to improve performance
  - support for all solid primitives
  - implementation of a complete demonstrator with AdePT and realistic geometry
- Implementation of gamma- and lepto-nuclear processes handling in AdePT
  - passing particles to Geant4 for final state generation
- Assessment of a possible implementation of specific processes such as optical photon propagation, neutron simulation
- Possible extension of AdePT to other sub-detectors
  - implementation of scoring mechanism for other types of detectors than calorimeters

Reconstruction

Phase 1: Track reconstruction - Activity areas

CPU

ACTS toolkit with several clients ATLAS, sPHENIX, FASER, LDMX

- full tracking chain available
- interfaced to DD4Hep and EDM4Hep

Published in CSBS, upcoming CHEP2023 talk (new EDM)

GPU

R&D Line of ACTS for combinatorial track Finding on GPUs (aka traccc project)

- divided into several sub projects
- symbiosis with EP R&D's phase 2 AdePT project in many areas

upcoming CHEP2023 talk (traccc)

OpenDataDetector

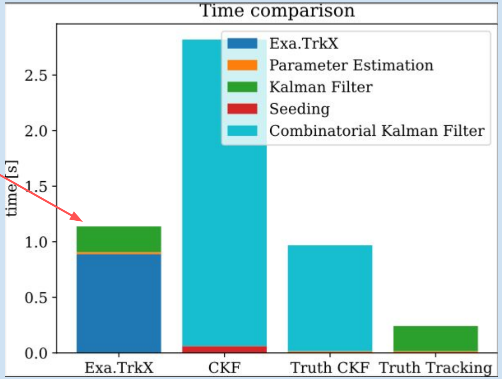
DD4Hep based (initially) Tracking detector for algorithmic and software R&D, upcoming CHEP2023 talk

acts - Highlight

https://github.com/acts-project/acts

Exa.TrkX + ACTS

GPU CPU

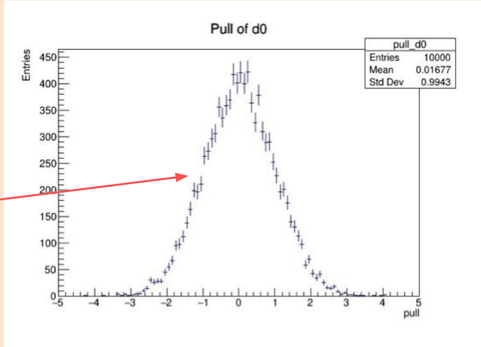


traccc - Highlight

https://github.com/acts-project/traccc

Kalman Filtering on CPU/GPU

(including first studies on thread divergence)



Open Data Detector - Highlights

https://gitlab.cern.ch/acts/OpenDataDetector/

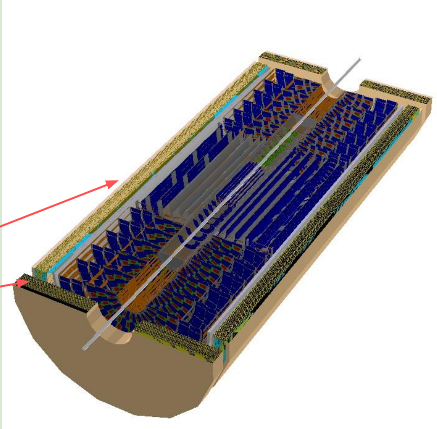
Tracking detector defined since quite a while

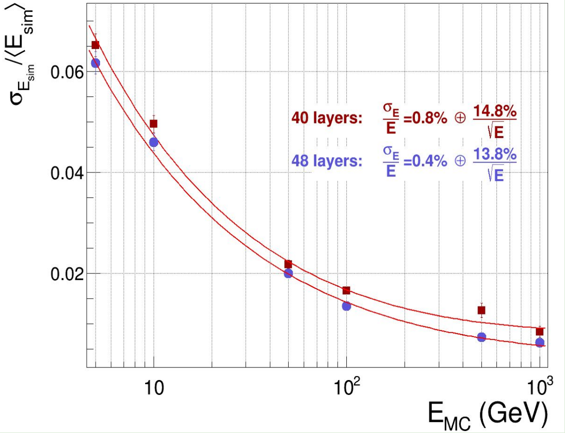
Dependency from ACTS decoupled

(recently updated/refined for AdePT project usage)

EM calorimeter added recently

HCAL + Muon system to be added




$\sigma_{E_{sim}}/\langle E_{sim} \rangle$
0.02
0.04
0.06
40 layers: $\frac{\sigma_E}{E}$ =0.8% ⊕ $\frac{14.8\%}{\sqrt{E}}$
48 layers: $\frac{\sigma_E}{E}$ =0.4% ⊕ $\frac{13.8\%}{\sqrt{E}}$
10
10^2
10^3
E_{MC} (GeV)

Phase 1: Reconstruction Calorimetry

https://github.com/key4hep/k4Clue

CPU&GPU READY

- **CLUE** (CLUstering of Energy) is a fast density-based clustering algorithm for the next generation of sampling calorimeter with high granularity in HEP
- Excellent performance shown for CMS HGCAL with both simulated and test beam data
- CLUE was adapted to run in the key4hep framework for full-layout detectors
  - EDM4hep I/O
  - Extend search algorithm to cylindrical surface introducing (r, φ) coordinates
  - **Include possibility to test several different calorimeter layouts**

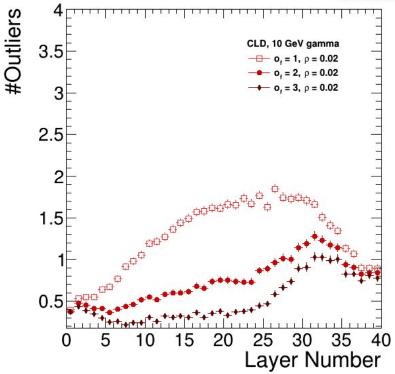
CPU&GPU READY

- Novel pattern recognition algorithm in CMS HGCAL to build showers (tracksters) in CMSSW
- The clustering logic is the same as in CLUE
- It groups Calorimeter Clusters belonging to different layers using projective coordinates

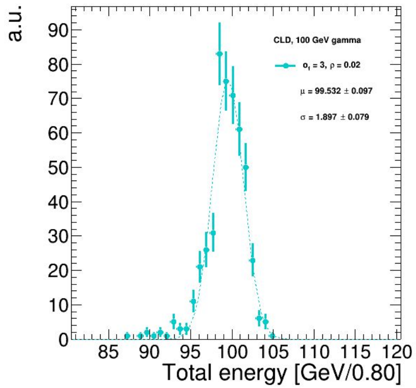
$$\left(\frac{r}{z} z_r, \frac{r}{z} z_r \phi\right)$$

- In 2022 CLUE3D was upgraded to official pattern reco algo for CMS/HGCAL

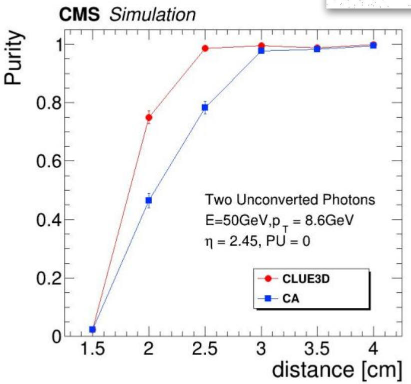
Phase 1: Reconstruction Calorimetry



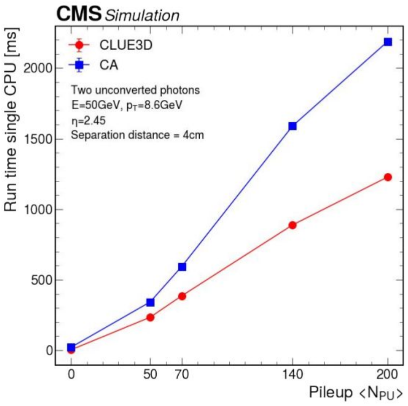
Example of parameter tuning for the CLD detector proposed for the FCC-ee collider in the case of events of single unconverted photon with 10 GeV. The **clusterized energy and the outlier and seed profiles** were used to determined the final CLUE input configuration.



Total energy clusterized using k4Clue using events of single unconverted photons with 100 GeV. The photons were generated using the CLD detector proposed for the FCC-ee collider but **similar results** were also obtained with the CLIC detector proposed for the CLIC collider.



Improvements with respect to the CA are **visible** for the trackster purity in the case of two unconverted photons with 50 GeV, separated by various distances, at the front face of the HGCAL.



Timing performance of CLUE3D vs. Cellular Automaton on a single core CPU, for the case of two unconverted photons. The run time of CLUE3D scales better with the pile-up with respect to the CA, **around 2 times faster** at 200 PU.

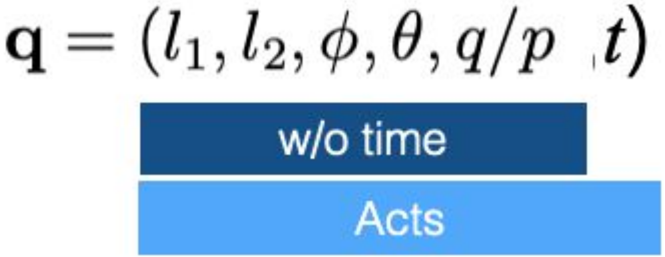
Phase 2: reconstruction with time information

- Event reconstruction in p-p collisions at high luminosity (HL-LHC, HE-LHC and FCC-hh with a pile-up(PU) of 200 and more), or in a high multiplicity heavy ion environment, suffers substantially from increased event complexity
- The spatial overlap of tracks and energy deposits from the additional collisions (PU) can severely *degrade the identification and the reconstruction of the hard interaction* and *can increase the rate of false triggers*.
- By associating tracks from a vertex and calorimetric deposits to hits and their corresponding times, the time at which the collision vertex or shower occurred can be reconstructed
- **Novel reconstruction techniques are needed to accomplish these tasks.**
- Requiring the compatibility between the measured time of tracks and showers will provide a powerful tool to *assist pile-up rejection*, *improve jets and p^{miss}_{T} resolution,* and to *better identify the primary vertex* of the triggered interaction.

Phase 2: track reconstruction with time information

ACTS toolkit has already expanded the track parameterisation with time information

- EDM with time information available
- Propagation, track fitting should use time information transparently like any other parameter



$$\mathbf{q} = (l_1, l_2, \phi, \theta, q/p, t)$$

Work package for phase 2 R&D&D(eployment)

- Develop & complete mathematical framework
  - Validate with Open Data Detector with time measurements
- Include time measurements in pattern recognition, (Combinatorial) Kalman Filtering
  - demonstrate performance in very high pile-up scenarios
  - demonstrate potential as time of flight tagger
- Work on existing ML based track reconstruction solutions to include time information
  - E.g. inclusion in Exa.TrkX pipeline as additional feature
- Include tracks with time measurements/stamps in vertex reconstruction
- Establish a functional framework for trackers with time measurements in ACTS

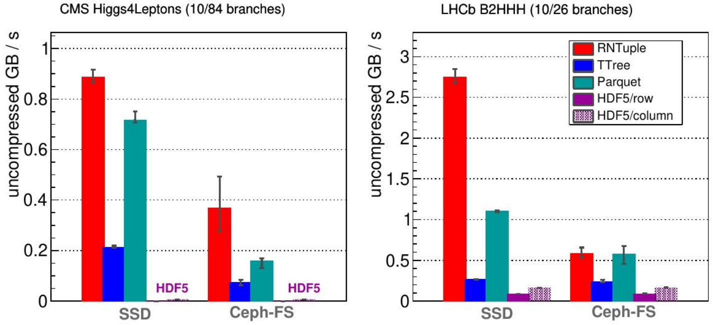
Phase 2: calorimetry reconstruction with time information

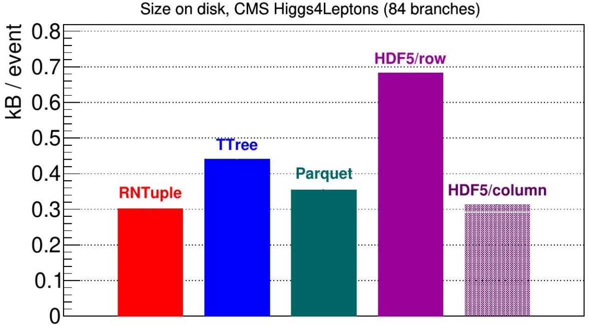
- Phase 1 Calorimetry project established a solid base
  - CLUE/K4CLUE developed and deployed (**CPU and GPU**)
  - CLU3D integration and deployment foreseen in 2023
- Precision timing for calorimetry is a major addition for HL-LHC and future colliders
  - it will provide a *powerful tool* to assist pile-up rejection, Jet and MET resolution, and particle isolation.
- Work package for phase 2 R&D:
  - Reviewing existing clustering and pattern recognition algorithms *to accommodate the usage of timing* information in calorimeters.
  - Include timing into clustering and pattern recognition algorithm inside calorimeters
    - use timing within calorimeters
    - exploit timing information from tracks
  - Use *performance portability* library for CPU and heterogeneous devices (GPUs and more).
  - Combination of timing information for neutral and charged particles for an optimal particle flow event interpretation (*eventually using advanced ML techniques*).

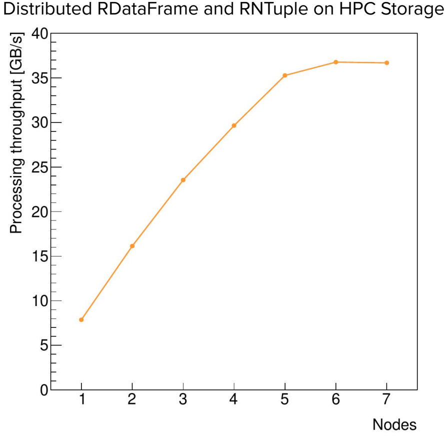
Analysis

Efficient Analysis: Results from Phase I

- RNTuple performance validation
  - Substantially better throughput and smaller files than TTree & other tools
- File-less storage: use of HPC object stores
  - ~35GB/s end-to-end throughput on 7-node cluster
  - Foundation for cloud object store support
  - Collaboration with openlab and Intel, Hewlett Packard Enterprise
- RNTuple nanoAOD writer in CMSSW
  - Collaboration with IRIS-HEP
- Inception of Distributed RDataFrame
  - Scale-out analysis framework
  - Python first, leverages Big Data tools and connects them to ROOT application knowledge







Efficient Analysis: Phase II Proposal

- Efficiently support analysis workflows **at scale**
  - Scale w.r.t. **EDM complexity**: Validation of RNTuple format and API with experiment data products and workflows
    - Ongoing work with CMS and ATLAS
    - Many non-obvious requirements, e.g. late schema extension, multi-threaded writes, I/O with custom memory allocators

  - Scale w.r.t. to **dataset versatility**: Facilitate reduction of user-generated redundancy through more robust and flexible “friends” and “chains”
    - Flexible means to combine existing data sets in new, virtual data sets
      - Persistified cuts in the form of masked columns
      - Re-shaping of existing data (e.g., on-the-fly “objectification”)
      - Combination of columns from different files / data sets
    - RNTuple metadata to track
      - data provenance (e.g., file derivation chain)
      - column dependencies and operators (e.g., varied columns, scale factors)
      - Standardized in file format to make information readily available to analysis tools

Efficient Analysis: Phase II Proposal

- Efficiently support analysis workflows **at scale**
  - **Scale-out** to HPC, scientific clouds, analysis facilities
    - S3 object store support in RNTuple
    - Investigation of the impact of RNTuple data on HEP data storage systems (XRootD, XCache)
    - Explore Distributed RDataFrame on different HPC infrastructures

- R&D programme foresees collaboration with experiments and infrastructure providers
  - Successful experience from phase I R&D
  - Discussions ongoing with experiments, CERN IT, Rucio

HEP Core Libraries (New Task)

Acceleration Core Lib

- Community moves code to GPU, even more so by 2028
    - Already multiple implementations of some common ingredients
- Expect significant benefit from centralising R&D and optimisation for multiple accelerator implementations (CUDA, SYCL and others)
    - 4-vector / linear algebra - *performance evaluation of CMS track finding with accelerated matrix operations*
- R&D on CPU histogram library and GPU filling (=data reduction) as part of HEP ecosystem
    - fitting, I/O, graphics, simple interfaces - *adaptation of ALICE O2 analysis framework histogram manager to new histograms*
- Significant demand from community, for years, without suitable alternative solution; yet impossible to absorb required R&D in regular program

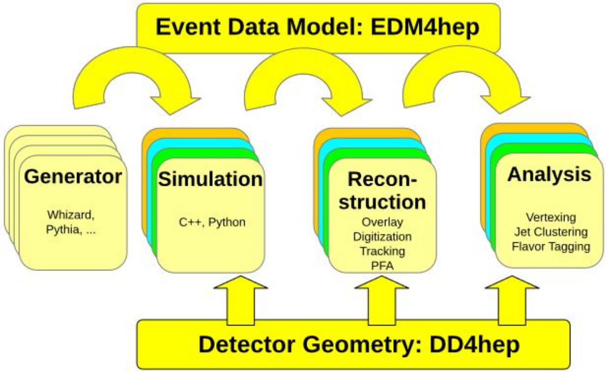
Language Interoperability Core Lib

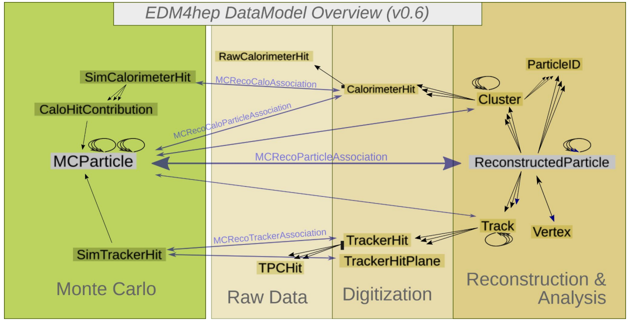
- Expectation: Python is here to stay. As is C++
  - We need interoperability, more than ever, but maintaining *performance*!
  - PyROOT is HEP's answer to this; it uses cppyy and ROOT's interpreter cling.
- Ingredients
  - Cppyy upstream has been redesigned; need to redesign PyROOT to benefit from it
  - LLVM compiler experts have invented C++ interpreter based on cling; need to re-design cling (alongside ROOT I/O and PyROOT) to benefit from that
- Significant R&D due to many complex ingredients at the foundation of HEP (I/O, language binding). Benefits: much improved C++ support; performance; simpler upgrades to newer C++ standards.

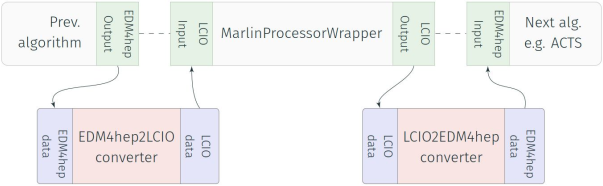
Turnkey Software → Heterogeneous Frameworks and New Languages

Phase I: achievements

- Integrated FCC and CLIC frameworks into common software stack
    - Based on: Gaudi, DD4hep, EDM4hep/podio, Geant4, ROOT, spack, etc.
    - Developed k4MarlinWrapper to integrate all iLCSoft processors
    - Developed EDM4hep event data model based in LCIO and FCC-edm, and evolved from there
- Build up an international community with participants from CEPC, CLIC, EIC, FCC, ILC, MC from CERN, DESY, IHEP, INFN, et al.
- Endorsed for ECFA Higgs/EWK/Top factories studies







Phase II: Heterogeneous Hardware / Language Evolution

- Based on Key4hep as common environment, prepare for the future of heterogeneous computing and efficient usage of HPCs
  - Research on combining multi-threading, multi-process and multi-node approaches
    - Development of an eventual implementation in a later stage
  - R&D on how to support multiple microarchitectures and GPUs efficiently
  - Can draw from gathered experience of CMS and LHCb in the usage of GPUs at trigger level
- Efficient data representation is essential for heterogeneous computing
  - Continue work on EDM4hep/PODIO and adapt it to new RNTuple developments
  - Investigation of data representations for various resources
- Stay open to recent rise of new programming languages like Julia or Rust
  - Future proofing for eventual shift of paradigms in the big data landscape
  - Forward looking effort to check interoperability of HEP software (EDM4hep) in multi-language environments

Summary

Resource Summary

- Resources dominated by personnel costs
- Balance student and fellow efforts
    - Talent pipeline
- Build links between software experts in EP groups
- Hardware allows shared diversity of platforms
- *Some additional descoping options have been discussed*

	Fellows (FTE)	Students (FTE)	Hardware (kCHF)
Year 1	1.5	6	125
Year 2	5.5	7	10
Year 3	4.5	9	0
Year 4	6	5	50
Year 5	6	4	0
Average	4.7	6.2	
Total			185

Grand Total ~4MCHF

Conclusions

- An ambitious programme to support software R&D is proposed
  - Working with other departments, labs and projects
- Tackling key challenges for future experiments
  - Including LHC upgrades
- Potential impacts are very high
  - Significant ultimate savings in human and material costs
    - Efficiencies (especially at LHC scales) bring great resource savings
    - Solve problems once, apply to many detectors
  - Maximise physics reach
    - State of the art techniques

Backup

Requested resources: Simulation

- Requested mix of students and fellows for both projects

- Timeline Takes into account extensions of Phase I fellows (fast sim sub-task)

Goods & consumables (kCHF)	Explanations
40	GPUs for heavy ML training and for AdePT development

(FTE)	Fellows	Students	Staff* for supervision
year 1	0.5	1	1.75
year 2	2.5	1	1.75
year 3	1.5	2	1.75
year 4	1	1	1.75
year 5	1	1	1.75

Requested resources: Reconstruction

- We tried to find the right balance between (doctoral) students and fellows.
- Hardware for testing merged/shared centrally among other sub-projects (60kCHF)

Year	**Tracking**		**Calorimetry**		**Total**		
	Students	**Fellows**	**Students**	**Fellows**	**Students**	**Fellows**	**Staff**
2024			1		1		0.6
2025		1	1		1	1	0.6
2026	1	1	2		3	1	0.6
2027	1		1	1	2	1	0.6
2028	1		1	1	2	1	0.6

Requested resources: Analysis

Goods & consumables (kCHF)	Explanations
20	Use of public cloud resources (through CERN broker)
20	Storage system for analysis benchmarking

(FTE)	Fellows	Students	Staff for supervision
year 1	0	1	0.5
year 2	1	1	0.5
year 3	1	1	0.5
year 4	1	1	0.5
year 5	1	1	0.5

- Same funding level as phase I
- Fellow hand-over phase I ➡ phase II in 2023

Requested Resources: HEP Core Libraries

ROOT can then "add" technical students and community contributions on top of the 0.6 students requested; but relies on fellows to supervise and steer R&D!

Goods & consumables (kCHF)	Explanations
30	Computer(s) with accelerators: NVIDIA + AMD GPUs, possibly RISC-V co-processor

(FTE)	Fellows	Students	Staff* for supervision
year 1	1	1	0.4
year 2	1	1	0.4
year 3	1	1	0.4
year 4	2		0.4
year 5	2		0.4

Requested Resources: Frameworks

Goods & consumables (kCHF)	Explanations
15	Testbed machine for GPU support in Gaudi

(FTE)	Fellows	Students	Staff* for supervision
year 1	1	2	1
year 2	1	2	1
year 3	1	2	1
year 4	2	1	1
year 5	2	1	1