WP7 Software

Team WP7: Graeme Stewart (WPL), André Sailer (DWPL), Juan Miguel Carceller, Swathi Sasikumar, Leonhard Reichenbach, Benedikt Hegner, Mateusz Fila, Andi Salzburger, Paul Gessinger-Befurt, Marco Rovere, Felice Pantaleo, Erica Brondolin, Aurora Perego, Anna Zaborowska, Peter McKeown, Piyush Raikwar, Michael Duehrssen-Debling, Joshua Beirer, Witek Pokorski, Andrei Gheata, Severin Diederichs, Juan González Caminero, Jakob Blomer, Vincenzo Padulano, Florine De Geus, Danilo Piparo, Lorenzo Moneta
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
EP R&D Phase 1

- Address key components of future needs (HL-LHC and future experiments)
  - Faster Simulation
  - Tracking and Calorimeter Reconstruction
  - Faster Analysis

- Supporting Future Collider Studies with ‘best of breed’ components
  - Turnkey Software Stack - this is a testbed for other developments
Towards Phase 2

**Timescales**
- Software programme that still applies to the 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 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
- Core Libraries is a new line of work
- **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

**Partners**
- We work in collaboration with several non-CERN institutes and initatives (DESY, IJCLab, IHEP, IBM, openlab), also, in particular through AIDAinnova
Faster Simulation with ML
EP R&D Phase-1 outcomes

- **Par04** - Geant4 example for ML-based Fast Simulation
  - Detector-readout independent showers (released on Zenodo)
  - Training and conversion (ONNX, LWTNN, LibTorch) of ML model
  - Inference in Geant4 (putting the generated hits back)
  - Released in Geant4 11.2 - including GPU support
  - Adoption of some of these tools for Gaussino, LHCb

- **MetaHEP** - a model able to adapt quickly to new detectors

- Easy to use pipeline in Kubeflow with automatic hyperparameter optimization in Katib

- Co-organization of **CaloChallenge**, ML competition for encouraging development and better understanding of FastSim.
  - Datasets of varying complexity (including Par04)
  - Various metrics for benchmarking and objective comparison
EP R&D Phase-1 outcomes

- Contributions to developing **Open Data Detector (ODD)**
  - Realistic detector geometry for algorithmic research and development purposes
  - Updates presented at **ML4Jets2023**
- Development of CaloDiT, a transformer-based diffusion model
- **Proof of concept** on extending the models to more realistic geometries like FCCee (ALLEGRO, CLD) and ODD
- **Investigated** realistic limits on speedup using FastSim, $O(10^3)$
- Contributions to developing experiment-independent fast calorimeter simulation library from ATLAS FastCaloSim (AF3)
  - Replaced ATLAS particle transport with Geant4 transport
  - Decoupled most of FastCaloSim code from Athena software
For EP R&D Phase-2 - Ongoing and future work

- Development of a Foundation model for FastSim (CaloDiT)
  - Train once on multiple geometries and quickly adapt to new ones
  - In collaboration with IBM and Openlab

Future work

- Scaling CaloDiT in terms of dataset and model size
- Optimization of CaloDiT
- Inference in DD4hep for studying effects of reconstruction
- Testing models in ATLAS software
- Ongoing support for LHCb fast sim development


Pretraining on Par04 & ODD, and adaptation to FCCeeALLEGRO

250 epochs of training from scratch (blue) are required match 20 epochs of adaptation (red)

~12x less training time
Faster Simulation on GPUs
AdePT status

● Achieved the initial goals of the R&D
  ○ Understand usability of GPUs for general particle transport simulation, seeking for potential speed up and/or usage of available GPU resource for the HEP simulation
    ■ Prototype e+, e− and γ EM shower simulation on GPU, evolve to realistic use-cases
      ● Geometry: VecGeom library, Physics: G4HepEm library
  ○ Transport for EM particles working on GPUs for LHC-complexity geometries
    ○ Excellent physics agreement within statistical fluctuation
    ○ Reproducibility of the simulation achieved
● Full integration with Geant4 applications
  ○ Possible to plug AdePT into existing Geant4 applications with minimal extra code
    ■ reusing existing sensitive detector implementations
● Main bottleneck: geometry – being addressed now (see following slides)
Bounded surface modeling for improving GPU performance

- Portable GPU-friendly header library
  - Algorithmic part independent on the backend, compilable with any native/portability compiler
  - Headers templated on the precision type to allow for a single-precision mode

- Code simplification and GPU performance
  - No virtual calls, no recursions, more work-balanced
  - Better device occupancy and kernel coherence
  - Reducing divergence and register usage on the GPU
Status and plans

Surface model supports **all solids** required by the experiments

**Current priorities:**
- Debugging complex geometries (CMS, LHCb): **Overlaps** require relocations
- Implementing **Bounding Volume Hierarchy** accelerating structure

**Next:**
- Performance measurements, testing calorimeters (ATLAS EMEC, CMS HGCAL, LHCb ECAL)
- Lightweight portability layer (instead of pure CUDA)

Missing the overlapping entering surface leads to missing Box 2 entirely
Track Reconstruction
EP R&D Phase 1 - Modern track reconstruction

Established a strong, feature-rich CPU baseline of track reconstruction
[P. Gessinger-Befurt, M. Kiehn, A. Salzburger et al, CSBS2022]  
- detector agnostic yet customizable  
  clients: ATLAS, FASER, sPHENIX, ePIC, ...  
- high performant  
  physics performance, computing performance  
- extendable

EP R&D initiative was extremely helpful to develop and define common interfaces  
- Geometry (DD4hep, GeoModel)  
- EDM (EDM4hep, PODIO)
EP R&D Phase 1 - Outcome and further R&D

Commonly* developed **OpenDataDetector**


- to develop and showcase algorithms & performance

[ P. Gessinger-Befurt, A. Salzburger, et al, CTD2023 ]

EDM frontend/backend separation

- PODIO/EDM4hep demonstrators

[ P. Gessinger-Befurt, CHEP2023 ]

Geometry (**detray**) and Simulation support for GPU R&D line **traccc**

- Exchange with AdePT on GPU geometry

*with **EP R&D members** from the fast simulation activity
Towards Phase 2

Shift focus to GPU R&D & **improve integration into Key4hep for FCCee**

- New geometry model allows easier binding to DD4hep, EDM4hep converters in place

- Re-establish full track reconstruction chain with time information: CPU/GPU code

- Enhance detray geometry for Calorimeter
  Simplified geometry building from DD4hep

- Develop full parameter transport through calorimeter with dense material
  Input for combined (time-aware) particle flow algorithms

- Combine with Calorimeter Reconstruction task to support modern particle flow algorithms

Achieved quasi-identical material (shown) and magnetic field accuracy on CPU/GPU
Calorimeter Reconstruction
k4CLUE: CLUE integration in Key4hep

- **CLUE**: fast density-based *clustering algorithm* already in use in HGCAL reconstruction
  - uses **energy density** to establish seeds, outliers, and followers in 2D planes
  - **GPU-friendly**
- **Key4hep integration:**
  - adapted to the common event data model, **EDM4hep**
  - adapted to run on **full detector**
  - supports **different** type of **calorimeter** layouts
  - github CI and EDM4hep **Validation**

Brondolin E., Pantaleo F., Rovere M., [The k4Clue package: Empowering Future Collider Experiments with the CLUE Algorithm](#)
k4CLUE: performance results

Analysis on three different future calorimeters for single gamma events:
- similar performance with respect to other baseline algorithms
- **good performance** in presence of noise
- clear **advantage** in terms of **timing** performance
  - not dependent on the number of input hits
  - only CPU version used, GPU version would be even faster

Brondolin E., Pantaleo F., Rovere M., *k4Clue: the CLUE Algorithm for Future Collider Experiments*, CTD2023
Towards Phase 2

- Integrating the k4Clue developments in the kalos/CLUE generic library with GPU support
- Testing and porting to GPU CLUE3D (clustering across full calorimeter)

Explore new solution for the pattern recognition with timing:
- Explore the time information available in the FCC detectors
- Integrate time information in pattern recognition algorithms in calorimeters reconstruction in CMS and FCC
  - plan to adapt HGCAL algorithmic techniques in the context of detectors foreseen for future collider experiments
- In the long run, combination with tracker info coming from 6D Kalman filter to perform a time-aware global event description
Faster Analysis
Faster analysis: I/O and programming model

Phase I: Two-fold strategy to enable next-generation HEP analysis

- Rewrite the HEP I/O layer
  - **RNTuple**: the future experiment data format
  - **2-5x throughput speedup** on fast storage devices w.r.t. **TTree**
  - Best-suited for HEP requirements compared to industry-standard tools
  - File-based **and** object storage (i.e. HPC and cloud)
  - **20% storage savings** for ATLAS production!

- Enable scaling an analysis to thousands of cores
  - Distributed RDataFrame **runs everywhere**
  - One API that unifies all HEP analyses, from a laptop to a Grid site
  - Performance tested up to thousands of cores
Since last year: smooth I/O and analysis integration

- RNTuple gives more performance and smaller dataset sizes
  - At no cost for the final user

- 1:1 compatibility with RDataFrame analysis
  - The tool can automatically detect TTree/RNTuple datasets

- Natively parallel: multithreaded or distributed analysis
  - RDataFrame+RNTuple+SWAN (ACAT’24)
  - Running a standard community benchmark with multi TB dataset (AGC)
Looking at Phase 2

- Challenge: complex event matching of RNTuple datasets
  - Gather desiderata from real community use cases
  - Extend TTree compositions (chains, friends)
  - 1:1 or 1:N relationships, unaligned events in the datasets
  - Provide API for higher-level tools
  - Ensure similar performance for most use cases
  - Vertical concatenations require clear definition of assumptions
  - Potential for significant storage savings by not duplicating columns into new files

- Validation at realistic scales
  - Also in collaboration with the community (CERN IT, experiment frameworks)
  - Test and benchmark on various HPC sites

- Future-proof the data format
  - Schema evolution, backwards and forwards compatibility features
HEP Core Libraries (New Task)
Core Libraries

● Future experiments’ will have huge and complex data sets
  ○ This requires a new platform of reengineered HEP foundation libraries and interfaces
    ■ Take advantage of the increasingly wide variety of modern computing devices, ergonomically
  ○ Not adapting to these trends is a considerable risk, which we mitigate with this work

● Histogramming on GPUs
  ○ Develop a new accelerator friendly histogram package for HEP, offering modern interfaces
  ○ Start by investigating ways to implement efficient data accumulation on accelerators and ensure that this functionality is interoperable with existing ROOT histogramming
  ○ The development of this new library will leverage the experience of ROOT’s and experiments’ existing implementations
Core Libraries

- **Sustainable C++-Python interoperability**
  - Current adoption trends suggest that in the 2030s most analyses will be written in Python
  - To keep usage of HEP's highly efficient C++ libraries high, effort will be invested in HEP's critical and unique Python binding, PyROOT
  - PyROOT has existed for many years within the ROOT package, it is very powerful since it is based on the Cling C++ interpreter (in the LLVM compiler suite)
    - However, needs to become more sustainable
  - Start by investigating new kinds of bindings to complement or replace the existing ones which are created dynamically
  - Assess the feasibility of upstreaming parts of cling/PyROOT to LLVM, thereby sharing their support cost with the community
  - Identify an ergonomic way to provide the new histograms developments in Python
Turnkey Software
Key4hep: Turnkey Software Stack

- Developed the complete turnkey software stack from simulation to analysis
  - Integrated FCC and CLIC frameworks into common software stack
  - Crucial for FCC feasibility study detector studies
- Successfully build up an international community with participants from CEPC, CLIC, EIC, FCC, ILC, Muon Collider from CERN, DESY, IHEP, INFN, et al.
  - Endorsed for ECFA Higgs/EWK/Top factories studies
- Key4hep has become an established platform with increased adoption from developers in the contributing projects

EP R&D in Key4hep

- **Crucial contributions from EP R&D**
  - Key4hep stack (nightly builds and stable releases) with over 500 packages on cvmfs
  - Integration with Gaudi for running reconstruction algorithms
    - Ongoing developments: tracking with ACTS, Pandora particle flow, overlay and more
  - In-memory dynamic conversion between LCIO (used by the ILC community) and EDM4hep, allowing algorithms that use either EDM4hep or LCIO to work together
  - Full reconstruction pipeline used as validation for detector design and development
  - Developments in podio and EDM4hep, like RNTuple support
  - Continuous validation system, improvements to software development

- **Outlook for Key4hep**
  - Consolidate and finish a stable version of EDM4hep
  - Participate in and benefit from the FCC studies
    - Increase in activity already happening
  - Needs continued input and sustained effort to continue to support the production mode
Heterogeneous Frameworks
Scheduling with heterogeneous resources

- Heterogeneous resources becoming more and more available and increasingly important for HPC
- Concurrent schedulers adopted by the experiments a decade ago
  - Main focus on multi-process and thread-parallel execution
  - Successful ad-hoc solutions for multi-node setups, non-blocking tasks and computation offloading
- New 3rd party libraries, frameworks and computing models for heterogeneous scheduling available
- Prototype new heterogeneous schedulers for future experiments
  - While learning and incorporating prior experience of CMSSW and Athena
Heterogeneous scheduling status and plans

Phase I - now
- **Done**: Extracted information about workloads used by the experiments:
  - Data flow and control flow graphs
  - Algorithm timings and now as well data object memory footprints
  - Presented at Gaudi Developer Meeting
- **Ongoing**: Prototype single-node scheduler with asynchronous offloading
  - Create demonstrators running mockup workloads
  - Started to evaluate the state-of-the-art task-graph heterogeneous scheduling libraries and different multi-tasking styles (cooperative scheduling)
- **Ongoing**: Investigate scheduling in new programming languages:
  - Prepare demonstrator in Julia using Dagger.jl framework, repository
  - Working with an Ukrainian Remote Student and soon with an additional Summer Student

Phase II - longer-term
- Distributed scheduling with multiple nodes and architectures
- Study energy-efficient scheduling with heterogeneous nodes
- Implementation strategies for next generation frameworks
Summary and Conclusions
EP Software R&D Summary

● Phase 1 of the Software Work Package of EP R&D achieved a great deal
  ○ Working high-performance software stack supporting FCC studies and used by many future experiment studies
  ○ R&D into advanced ML simulation techniques for calorimeters
  ○ State of the art reconstruction techniques for trackers and calorimeters
  ○ Development of world class performant file format for HEP, with ergonomic analysis interfaces

● We look forward to continuing these strong lines of development and adding new ones
  ○ Improved frameworks support for heterogeneous environments
  ○ Particle tracking on GPUs
  ○ Core library support for GPUs
  ○ Full incorporation of timing into reconstruction, with “full-event” understanding

● R&D with strong links to current and future experiments
  ○ Aim at early adoption into production as soon as possible
Backup
Adaptation using CaloDiT

FCCeeALLEGRO

250 epochs for training from scratch
20 epochs for adaptation

At 200K samples
~25x less training time
<50% of the data

Preliminary results
RNTuple: the path to production

- RNTuple workshop 2023 ([indico](#))
  - Participation of LHC core computing devs, CERN IT, HEP-CCE

- Work towards the release of a stable binary format
  - Allows stakeholders to battle test future data pipelines
  - RNTuple API review by external experts ([HEP-CCE](#))

- Integrating RNTuple in experiment production
  - All ATLAS data products available, most of CMS
  - +20% storage saving in ATLAS DAODs

---

**RNTuple: A Quick Look at DAOD Performance**

- Current studies indicate about 20% storage savings is possible in DAODs
  - It’s important to note TTree is heavily optimized over the last 20 years
  - Similar optimization studies will be carried out for RNTuple prior to production

---

ACAT ’24, S. Mete
VecGeom surface model targeting GPUs

Surface model now supports all solids used by the experiments - conversion time and memory footprint under control

<table>
<thead>
<tr>
<th></th>
<th># touchables [million]</th>
<th>conversion time [s]</th>
<th>memory [MB]</th>
</tr>
</thead>
<tbody>
<tr>
<td>cms_2018</td>
<td>2.1</td>
<td>3.0</td>
<td>102</td>
</tr>
<tr>
<td>cms_TB_HGCAL</td>
<td>0.06</td>
<td>0.5</td>
<td>28.4</td>
</tr>
<tr>
<td>cms_2026D98</td>
<td>13.1</td>
<td>18.0</td>
<td>278</td>
</tr>
<tr>
<td>LHCb_Upgrade</td>
<td>18.5</td>
<td>19.5</td>
<td>106</td>
</tr>
<tr>
<td>LHCb_ECal_HCal</td>
<td>18.4</td>
<td>0.3</td>
<td>4.4</td>
</tr>
<tr>
<td>ATLAS_EMEC</td>
<td>0.08</td>
<td>0.9</td>
<td>62.6</td>
</tr>
</tbody>
</table>