AIDA 2020

Advanced European Infrastructures

WP3 - Advanced Software Summary

Frank Gaede, Witek Pokorski

AIDA2020 Final Annual Meeting, 30.04.2020

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 654168.

AIDA 2020

Overview - WP3

Advanced simulation and reconstruction for HEP

- Core software
  - **DD4hep** and **USolids/VecGeom** extensions
  - alignment and conditions data
  - **EDM** toolkit and framework extensions
- Simulation
  - **DDG4**: Geant4 based simulation toolkit
- Reconstruction
  - advanced tracking tools
  - advanced particle flow algorithms

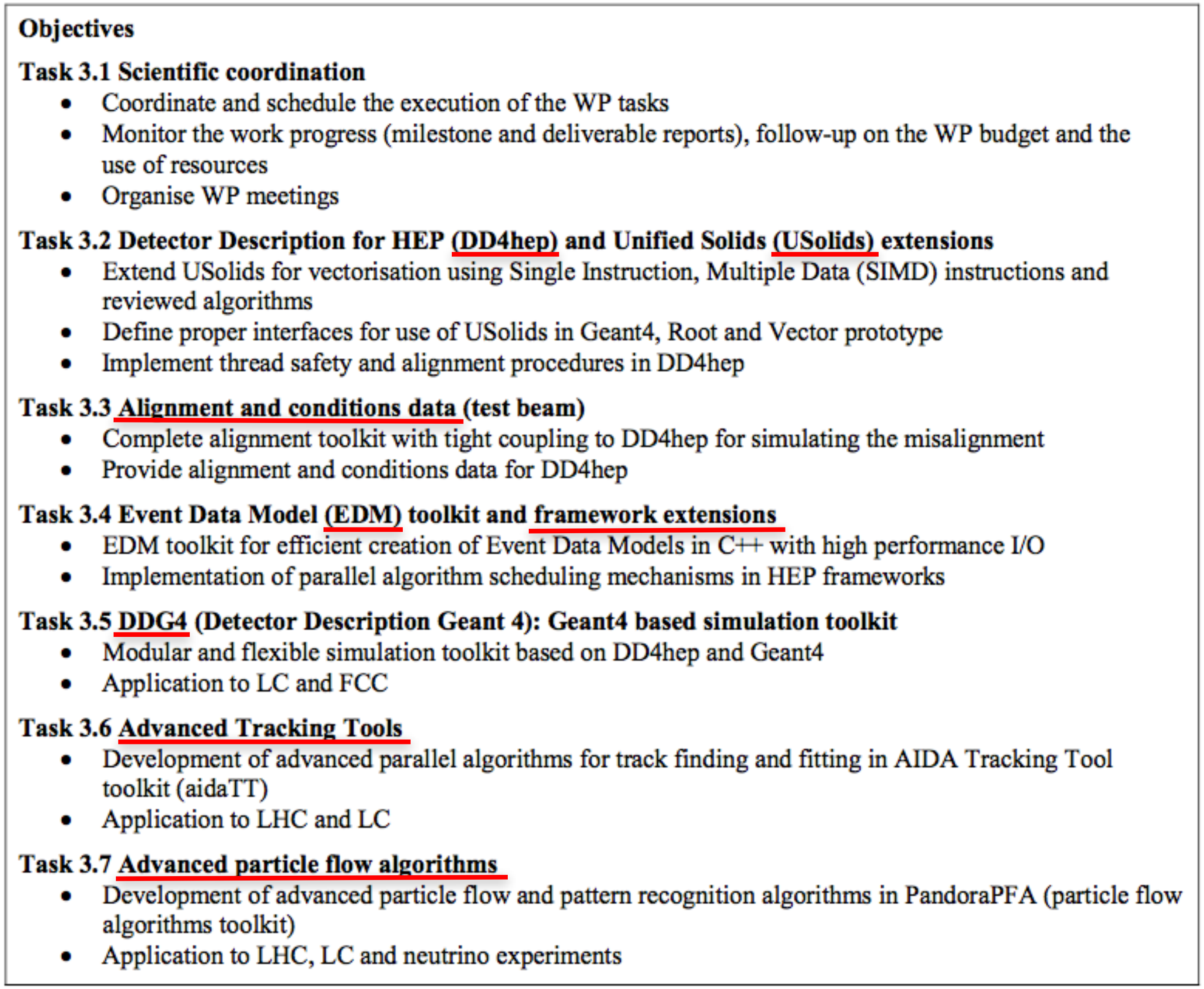
address high performance computing in all tasks: parallelization, vectorization

- Partners:
- **CERN, DESY, LAL, LLR, U-Manchester, U-Cambridge**

General Philosophy

- develop software that is needed for ongoing projects:
  - **LHC** upgrade, **ILC** and **CLIC**, **FCC** or neutrino
- focus on **generality** and **re-usability** already in design of the software
- tools should address the **current needs** and at the same time be **generic enough** to be applicable for other/**all HEP experiments**
- seek collaboration with **Hep Software Foundation (HSF)**
  - some packages are already official HSF projects

WP3 - Tasks



Objectives

Task 3.1 Scientific coordination

- Coordinate and schedule the execution of the WP tasks
- Monitor the work progress (milestone and deliverable reports), follow-up on the WP budget and the use of resources
- Organise WP meetings

Task 3.2 Detector Description for HEP (DD4hep) and Unified Solids (USolids) extensions

- Extend USolids for vectorisation using Single Instruction, Multiple Data (SIMD) instructions and reviewed algorithms
- Define proper interfaces for use of USolids in Geant4, Root and Vector prototype
- Implement thread safety and alignment procedures in DD4hep

Task 3.3 Alignment and conditions data (test beam)

- Complete alignment toolkit with tight coupling to DD4hep for simulating the misalignment
- Provide alignment and conditions data for DD4hep

Task 3.4 Event Data Model (EDM) toolkit and framework extensions

- EDM toolkit for efficient creation of Event Data Models in C++ with high performance I/O
- Implementation of parallel algorithm scheduling mechanisms in HEP frameworks

Task 3.5 DDG4 (Detector Description Geant 4): Geant4 based simulation toolkit

- Modular and flexible simulation toolkit based on DD4hep and Geant4
- Application to LC and FCC

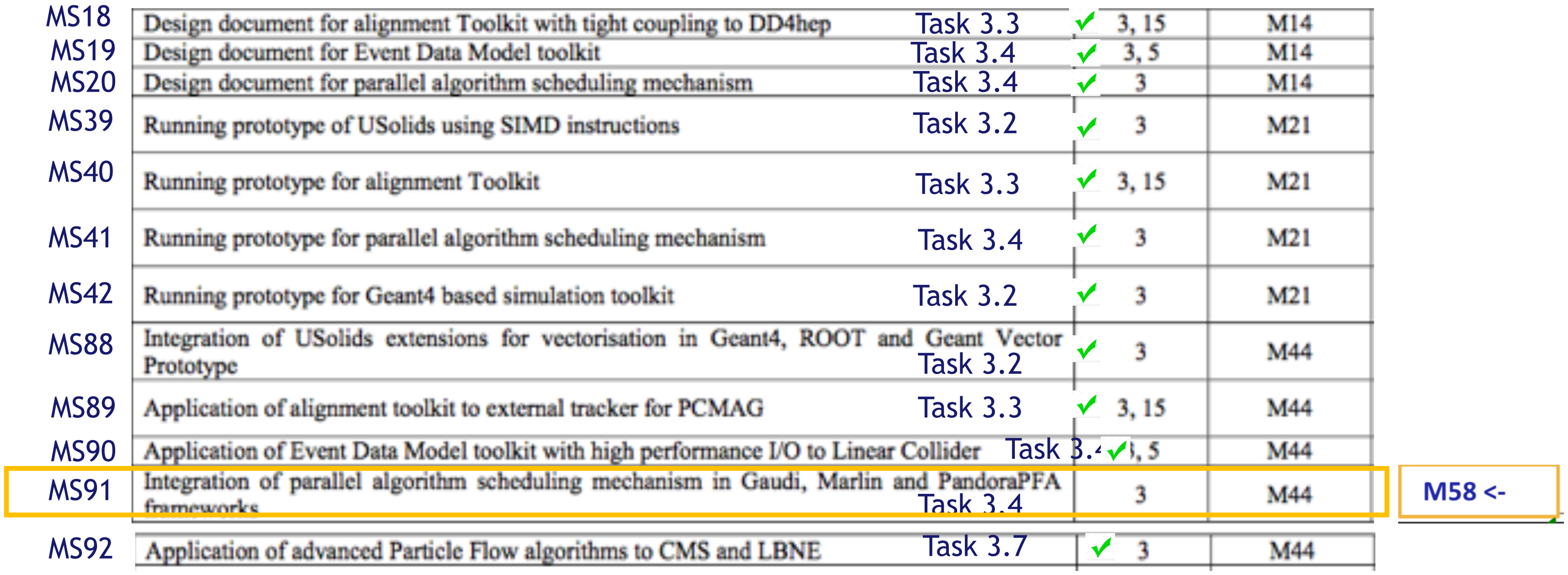
Task 3.6 Advanced Tracking Tools

- Development of advanced parallel algorithms for track finding and fitting in AIDA Tracking Tool toolkit (aidaTT)
- Application to LHC and LC

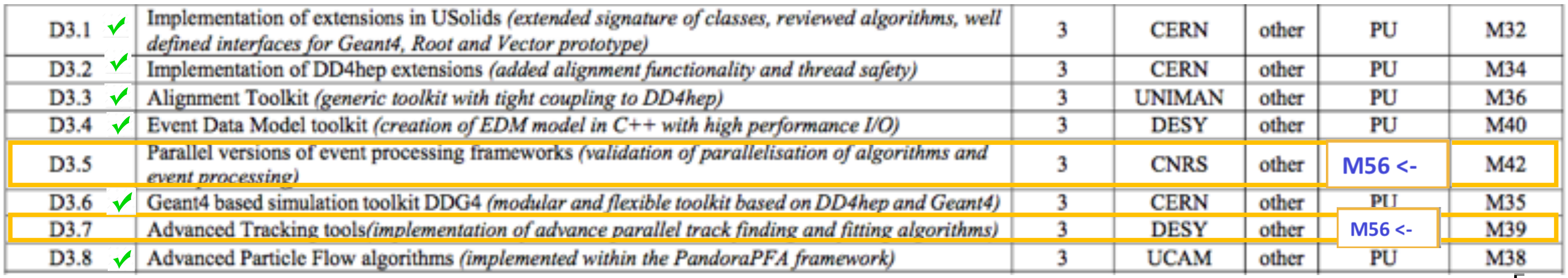
Task 3.7 Advanced particle flow algorithms

- Development of advanced particle flow and pattern recognition algorithms in PandoraPFA (particle flow algorithms toolkit)
- Application to LHC, LC and neutrino experiments

WP3 Milestones and Deliverables *in year 5*

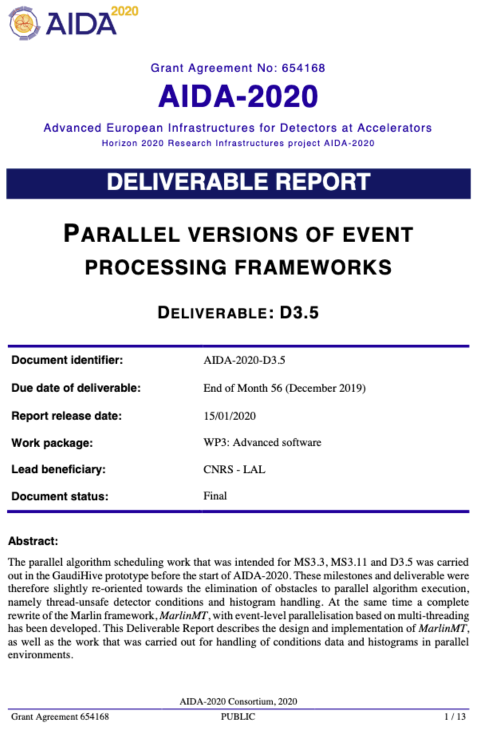


			Achieved on time			Postponed to Year 5
MS18	Design document for alignment Toolkit with tight coupling to DD4hep	Task 3.3	✓	3, 15	M14	
MS19	Design document for Event Data Model toolkit	Task 3.4	✓	3, 5	M14	
MS20	Design document for parallel algorithm scheduling mechanism	Task 3.4	✓	3	M14	
MS39	Running prototype of USolids using SIMD instructions	Task 3.2	✓	3	M21	
MS40	Running prototype for alignment Toolkit	Task 3.3	✓	3, 15	M21	
MS41	Running prototype for parallel algorithm scheduling mechanism	Task 3.4	✓	3	M21	
MS42	Running prototype for Geant4 based simulation toolkit	Task 3.2	✓	3	M21	
MS88	Integration of USolids extensions for vectorisation in Geant4, ROOT and Geant Vector Prototype	Task 3.2	✓	3	M44	
MS89	Application of alignment toolkit to external tracker for PCMAG	Task 3.3	✓	3, 15	M44	
MS90	Application of Event Data Model toolkit with high performance I/O to Linear Collider	Task 3.4	✓	3, 5	M44	
MS91	Integration of parallel algorithm scheduling mechanism in Gaudi, Marlin and PandoraPFA frameworks	Task 3.4		3	M44	M58 <-
MS92	Application of advanced Particle Flow algorithms to CMS and LBNE	Task 3.7	✓	3	M44	



D3.1	✓	Implementation of extensions in USolids *(extended signature of classes, reviewed algorithms, well defined interfaces for Geant4, Root and Vector prototype)*	3	CERN	other	PU	M32
D3.2	✓	Implementation of DD4hep extensions *(added alignment functionality and thread safety)*	3	CERN	other	PU	M34
D3.3	✓	Alignment Toolkit *(generic toolkit with tight coupling to DD4hep)*	3	UNIMAN	other	PU	M36
D3.4	✓	Event Data Model toolkit *(creation of EDM model in C++ with high performance I/O)*	3	DESY	other	PU	M40
D3.5		Parallel versions of event processing frameworks *(validation of parallelisation of algorithms and event processing)*	3	CNRS	other	M56 <-	M42
D3.6	✓	Geant4 based simulation toolkit DDG4 *(modular and flexible toolkit based on DD4hep and Geant4)*	3	CERN	other	PU	M35
D3.7		Advanced Tracking tools*(implementation of advance parallel track finding and fitting algorithms)*	3	DESY	other	M56 <-	M39
D3.8	✓	Advanced Particle Flow algorithms *(implemented within the PandoraPFA framework)*	3	UCAM	other	PU	M38

Final Deliverables/ milestones



Grant Agreement No: 654168

AIDA-2020

Advanced European Infrastructures for Detectors at Accelerators

Horizon 2020 Research Infrastructures project AIDA-2020

DELIVERABLE REPORT

PARALLEL VERSIONS OF EVENT PROCESSING FRAMEWORKS

DELIVERABLE: D3.5

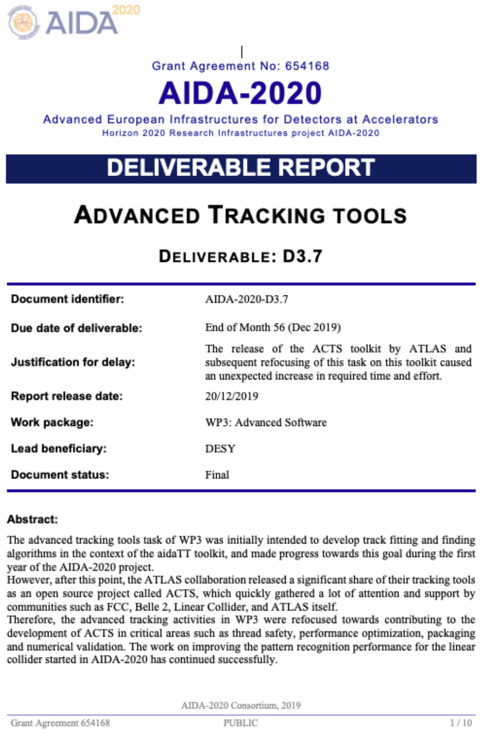
Document identifier:	AIDA-2020-D3.5
Due date of deliverable:	End of Month 56 (December 2019)
Report release date:	15/01/2020
Work package:	WP3: Advanced software
Lead beneficiary:	CNRS - LAL
Document status:	Final

Abstract:

The parallel algorithm scheduling work that was intended for MS3.3, MS3.11 and D3.5 was carried out in the GaudiHive prototype before the start of AIDA-2020. These milestones and deliverable were therefore slightly re-oriented towards the elimination of obstacles to parallel algorithm execution, namely thread-unsafe detector conditions and histogram handling. At the same time a complete rewrite of the Marlin framework, *MarlinMT*, with event-level parallelisation based on multi-threading has been developed. This Deliverable Report describes the design and implementation of *MarlinMT*, as well as the work that was carried out for handling of conditions data and histograms in parallel environments.

AIDA-2020 Consortium, 2020

Grant Agreement 654168 PUBLIC 1 / 13



Grant Agreement No: 654168

AIDA-2020

Advanced European Infrastructures for Detectors at Accelerators

Horizon 2020 Research Infrastructures project AIDA-2020

DELIVERABLE REPORT

ADVANCED TRACKING TOOLS

DELIVERABLE: D3.7

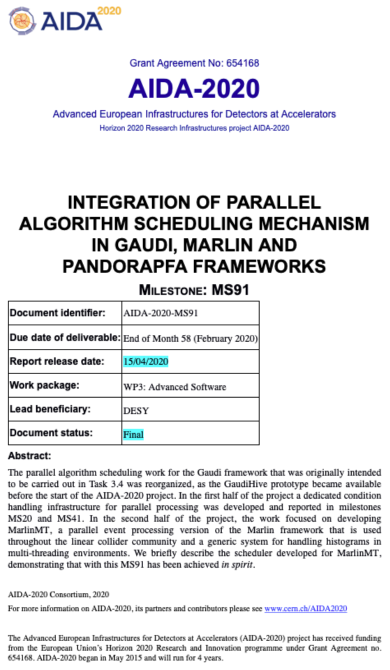
Document identifier:	AIDA-2020-D3.7
Due date of deliverable:	End of Month 56 (Dec 2019)
Justification for delay:	The release of the ACTS toolkit by ATLAS and subsequent refocusing of this task on this toolkit caused an unexpected increase in required time and effort.
Report release date:	20/12/2019
Work package:	WP3: Advanced Software
Lead beneficiary:	DESY
Document status:	Final

Abstract:

The advanced tracking tools task of WP3 was initially intended to develop track fitting and finding algorithms in the context of the aidaTT toolkit, and made progress towards this goal during the first year of the AIDA-2020 project.
However, after this point, the ATLAS collaboration released a significant share of their tracking tools as an open source project called ACTS, which quickly gathered a lot of attention and support by communities such as FCC, Belle 2, Linear Collider, and ATLAS itself.
Therefore, the advanced tracking activities in WP3 were refocused towards contributing to the development of ACTS in critical areas such as thread safety, performance optimization, packaging and numerical validation. The work on improving the pattern recognition performance for the linear collider started in AIDA-2020 has continued successfully.

AIDA-2020 Consortium, 2019

Grant Agreement 654168 PUBLIC 1 / 10



Grant Agreement No: 654168

AIDA-2020

Advanced European Infrastructures for Detectors at Accelerators

Horizon 2020 Research Infrastructures project AIDA-2020

INTEGRATION OF PARALLEL ALGORITHM SCHEDULING MECHANISM IN GAUDI, MARLIN AND PANDORAPFA FRAMEWORKS

MILESTONE: MS91

Document identifier:	AIDA-2020-MS91
Due date of deliverable:	End of Month 58 (February 2020)
Report release date:	15/04/2020
Work package:	WP3: Advanced Software
Lead beneficiary:	DESY
Document status:	Final

Abstract:

The parallel algorithm scheduling work for the Gaudi framework that was originally intended to be carried out in Task 3.4 was reorganized, as the GaudiHive prototype became available before the start of the AIDA-2020 project. In the first half of the project a dedicated condition handling infrastructure for parallel processing was developed and reported in milestones MS20 and MS41. In the second half of the project, the work focused on developing MarlinMT, a parallel event processing version of the Marlin framework that is used throughout the linear collider community and a generic system for handling histograms in multi-threading environments. We briefly describe the scheduler developed for MarlinMT, demonstrating that with this MS91 has been achieved *in spirit*.

AIDA-2020 Consortium, 2020

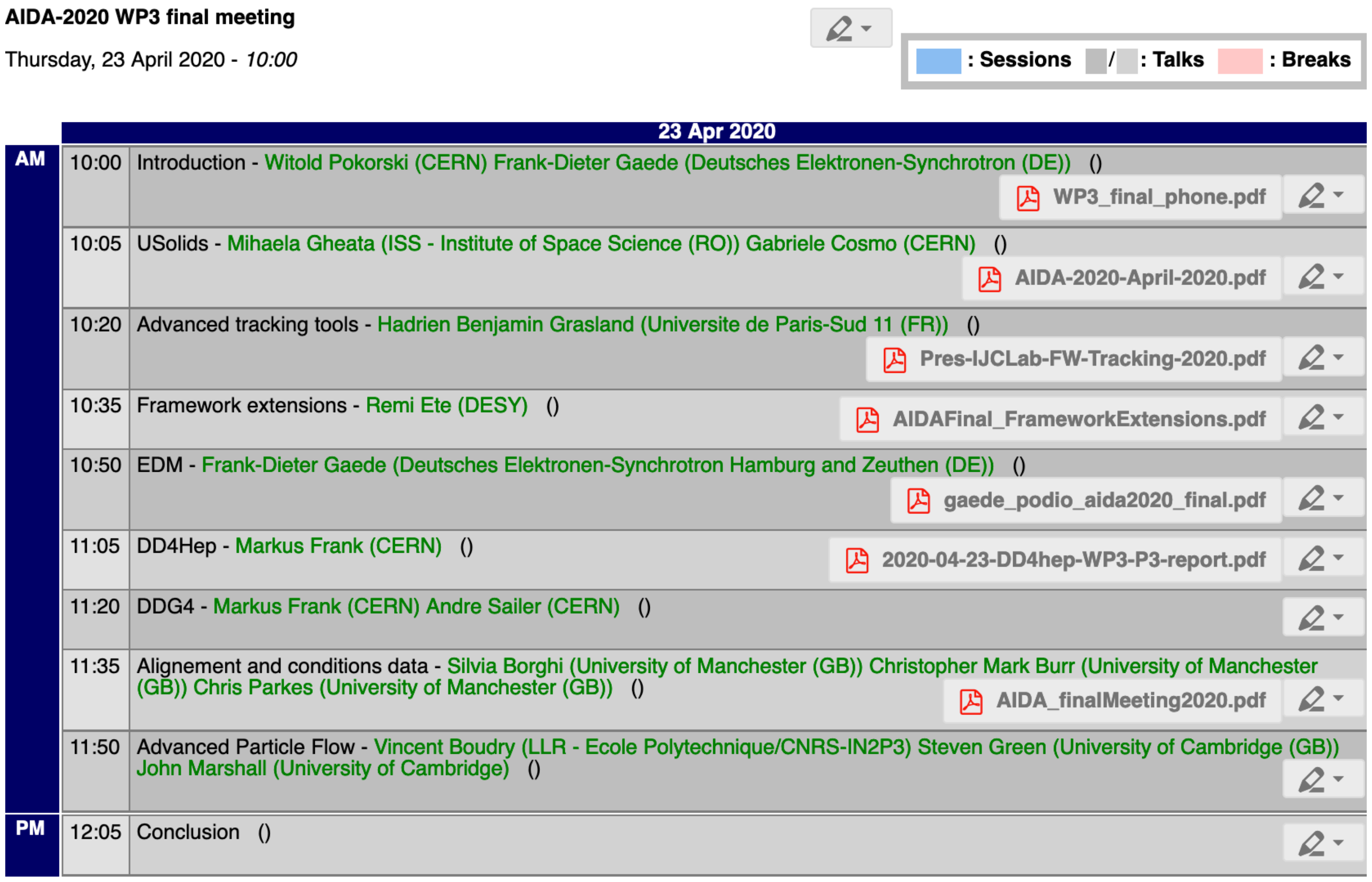
For more information on AIDA-2020, its partners and contributors please see www.cern.ch/AIDA2020

The Advanced European Infrastructures for Detectors at Accelerators (AIDA-2020) project has received funding from the European Union's Horizon 2020 Research and Innovation programme under Grant Agreement no. 654168. AIDA-2020 began in May 2015 and will run for 4 years.

- have submitted the final deliverables/milestones postponed to year 5 ~**on time:**
- **D3.5 Parallel Versions of Event Processing Frameworks**
- **D3.7 Advanced Tracking Tools**
- **M91: Integration of parallel processing algorithms in Gaudi, Marlin and PandoraPFA**

Final WP3 Meeting

- held on April 23 - find all task reports at:
- https://indico.cern.ch/event/910222/



AIDA-2020 WP3 final meeting

Thursday, 23 April 2020 - *10:00*

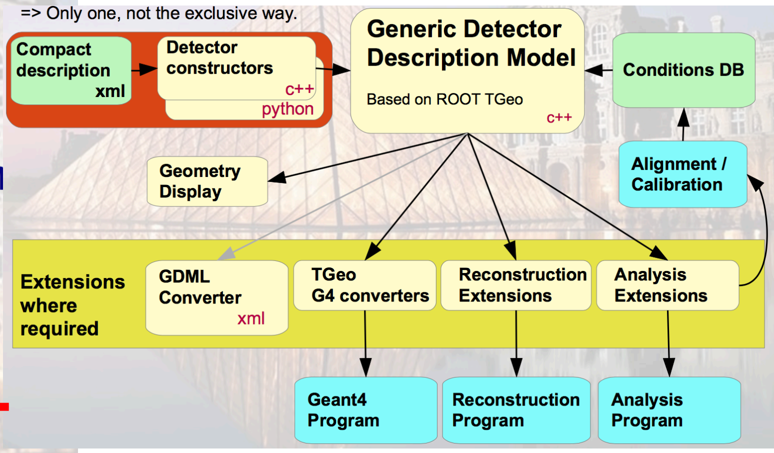
🟦 : Sessions ▨/▨ : Talks 🟥 : Breaks

		23 Apr 2020
AM	10:00	Introduction - Witold Pokorski (CERN) Frank-Dieter Gaede (Deutsches Elektronen-Synchrotron (DE)) () WP3_final_phone.pdf
	10:05	USolids - Mihaela Gheata (ISS - Institute of Space Science (RO)) Gabriele Cosmo (CERN) () AIDA-2020-April-2020.pdf
	10:20	Advanced tracking tools - Hadrien Benjamin Grasland (Universite de Paris-Sud 11 (FR)) () Pres-IJCLab-FW-Tracking-2020.pdf
	10:35	Framework extensions - Remi Ete (DESY) () AIDAFinal_FrameworkExtensions.pdf
	10:50	EDM - Frank-Dieter Gaede (Deutsches Elektronen-Synchrotron Hamburg and Zeuthen (DE)) () gaede_podio_aida2020_final.pdf
	11:05	DD4Hep - Markus Frank (CERN) () 2020-04-23-DD4hep-WP3-P3-report.pdf
	11:20	DDG4 - Markus Frank (CERN) Andre Sailer (CERN) ()
	11:35	Alignement and conditions data - Silvia Borghi (University of Manchester (GB)) Christopher Mark Burr (University of Manchester (GB)) Chris Parkes (University of Manchester (GB)) () AIDA_finalMeeting2020.pdf
	11:50	Advanced Particle Flow - Vincent Boudry (LLR - Ecole Polytechnique/CNRS-IN2P3) Steven Green (University of Cambridge (GB)) John Marshall (University of Cambridge) ()
PM	12:05	Conclusion ()

DD4hep - Overview

M.Frank

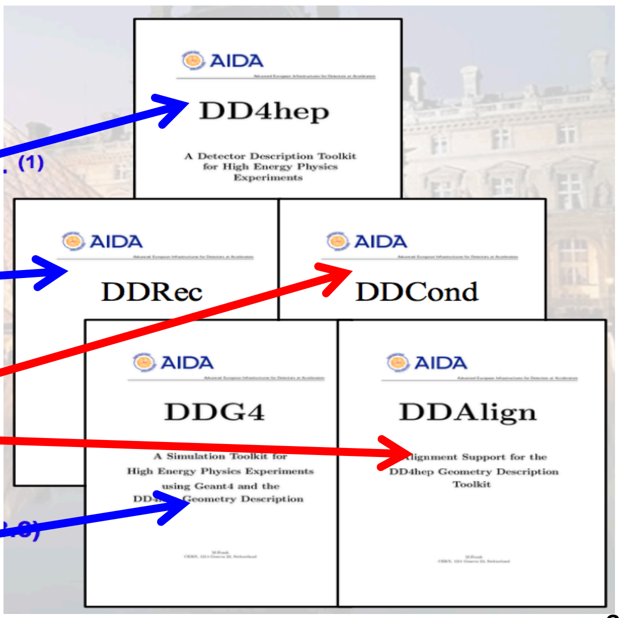
- **Develop a detector description**
  - **For the full experiment life cycle**
    - **detector concept development, optimization**
    - **detector construction and operation**
    - **“Anticipate the unforeseen”**
  - **Consistent description, with single source, which supports**
    - **simulation, reconstruction, analysis**
  - **Full description, including**
    - **Geometry, readout, alignment, calibration etc.**



core modules

latest developments

simulation using Geant4



AIDA

DD4hep

A Detector Description Toolkit for High Energy Physics Experiments

AIDA

DDRec

AIDA

DDCond

AIDA

DDG4

A Simulation Toolkit for High Energy Physics Experiments using Geant4 and the DD4hep Geometry Description

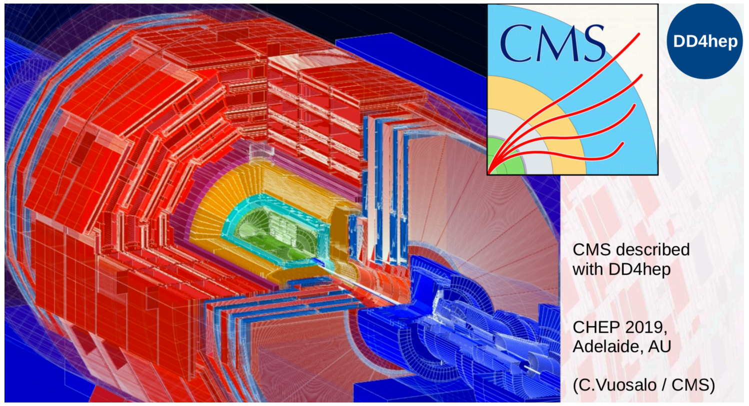
AIDA

DDAlign

Alignment Support for the DD4hep Geometry Description Toolkit

DD4hep - (New) Users

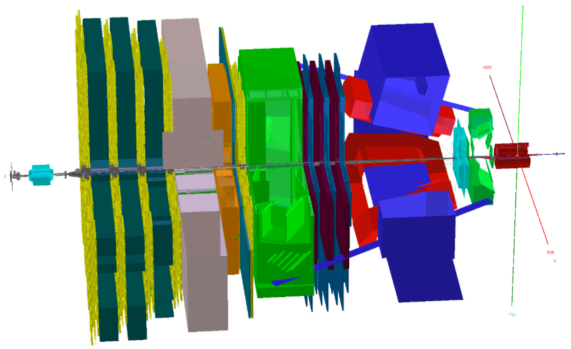
M.Frank



CMS described with DD4hep

CHEP 2019, Adelaide, AU

(C.Vuosalo / CMS)



DD4hep user base is growing - now used by all **future collider projects** and @ **LHC**

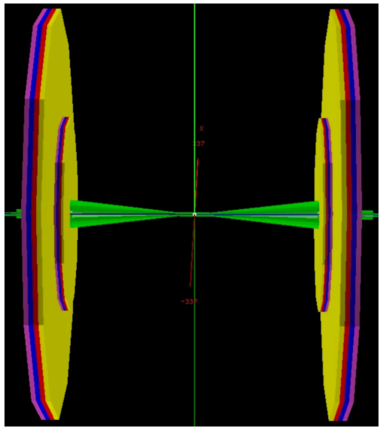
ILC	F. Gaede et al.
CLICdp	A. Sailer et al.
SiD	D. Protopopescu et al.
FCC-eh	P. Kostka et al.
FCC-hh	A. Salzburger et al.
FCC-ee	O. Viazlo (CLD design), N. Alipour, G. Voutsinas
SCTF	Super-Charm-Tau Factory designs (Novosibirsk, Bejing)
EIC	Evaluation considered/started (W. Armstrong et al.)
CEPC	Used for design studies (W. Li et al., IHEP)
LHCb	LHCb Upgrade for Run III (B.Couturier et al.)
CMS	Usage for upgrade - CHEP2020 (Y.Osborne et al.)
CALICE	Calorimeter R&D, started

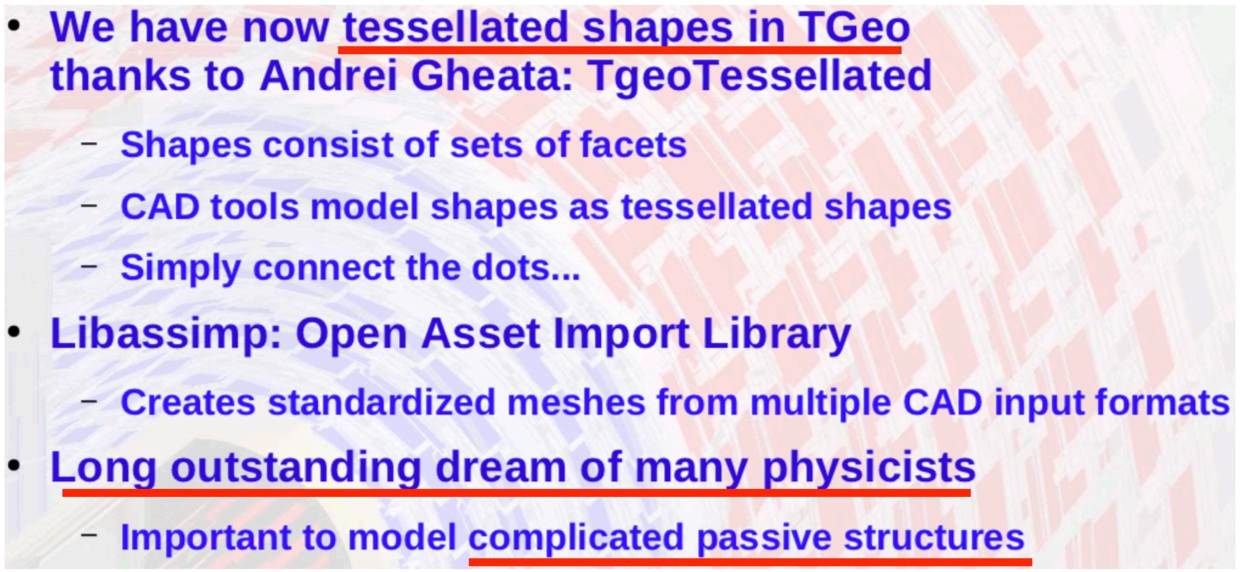
- adaptation of DD4hep by CMS and LHCb marks a **great success** of the project
- demonstrates that DD4hep can be used for very complex, existing detectors (the full life cycle)

DD4hep - New Features

M.Frank

- adaptation by new users also triggered quite some developments and improvements:
  - reflected (mirrored) volumes
  - additional shapes: cut tube 8-point solid, …
  - surfaces and optical properties
  - single system of units (G4/Root)





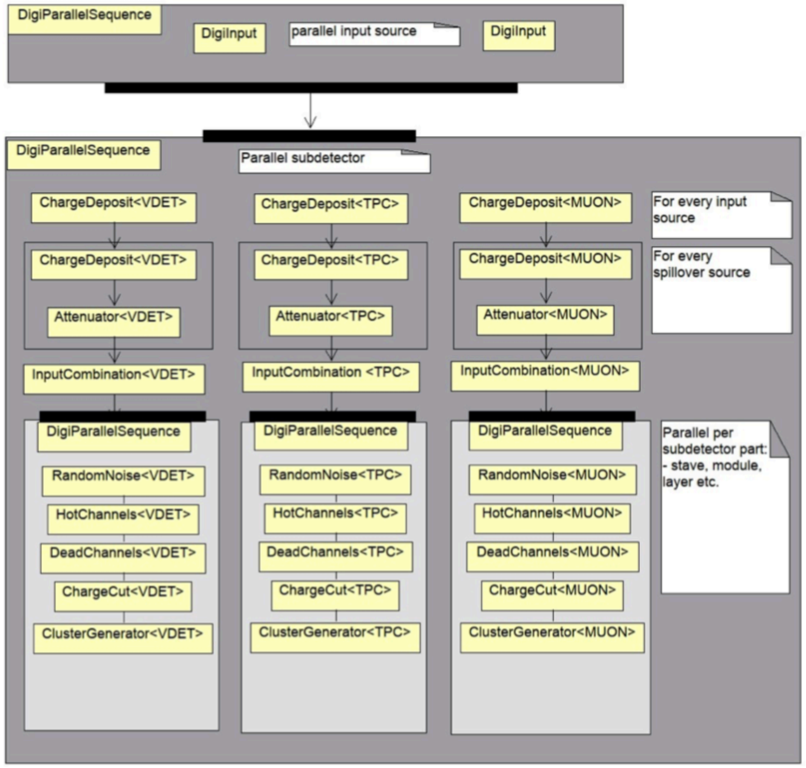
- We have now tessellated shapes in TGeo thanks to Andrei Gheata: TgeoTessellated
  - Shapes consist of sets of facets
  - CAD tools model shapes as tessellated shapes
  - Simply connect the dots…
- Libassimp: Open Asset Import Library
  - Creates standardized meshes from multiple CAD input formats
- Long outstanding dream of many physicists
  - Important to model complicated passive structures

DD4hep - Plans

M.Frank

- DDDigi: digitization component
  - conceptual design work has started …

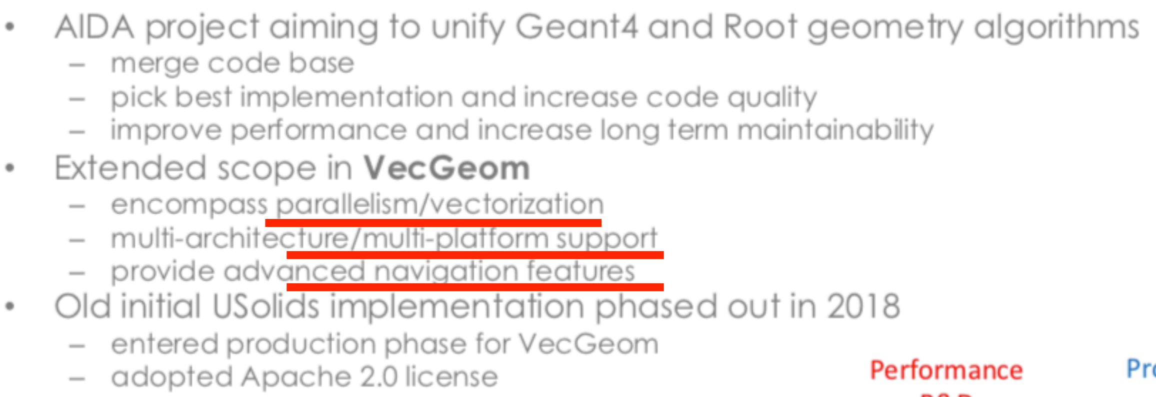
- **Handle collision overlays**
- **Handle event spillover**
  - **Process detector response from collisions earlier or later to the central event**
- **Noise handling**
  - **Random noise hits**
  - **Hot/dead channel emulation**
  - **Add noise to all channels with cut-off**



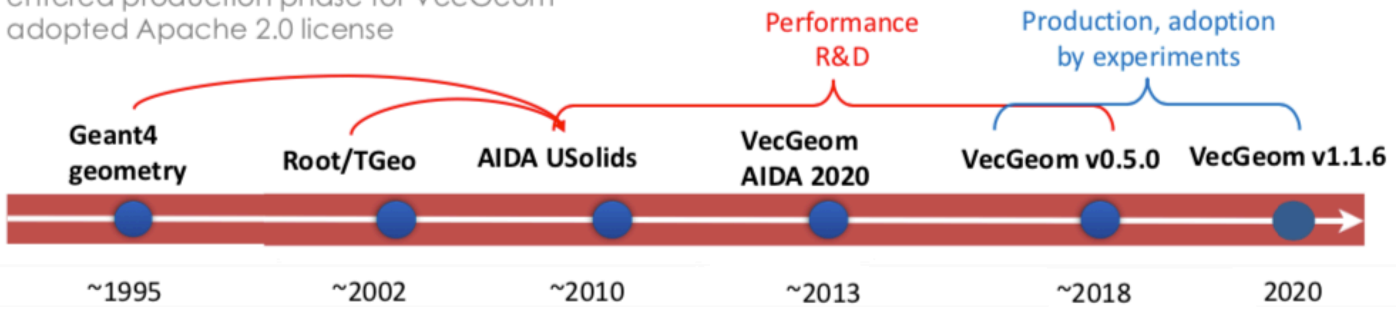
- work on DD4hep will continue after AIDA2020
- implementation of DDDigi planned for AIDAinnova

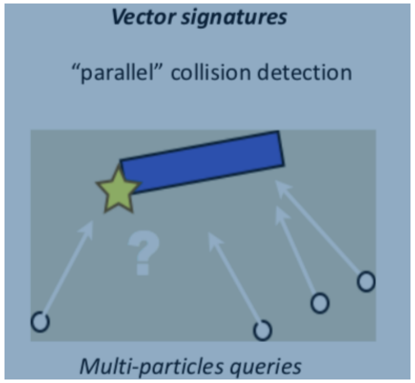
VecGeom - Overview

G.Cosmo

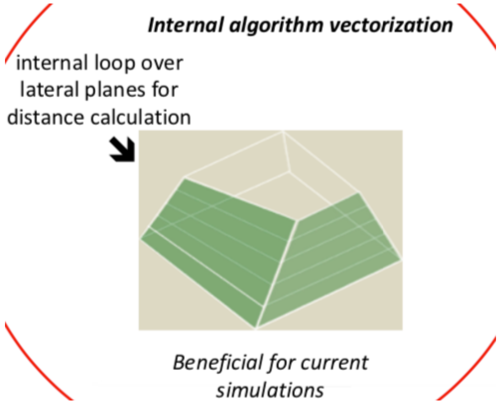


- AIDA project aiming to unify Geant4 and Root geometry algorithms
  - merge code base
  - pick best implementation and increase code quality
  - improve performance and increase long term maintainability
- Extended scope in **VecGeom**
  - encompass parallelism/vectorization
  - multi-architecture/multi-platform support
  - provide advanced navigation features
- Old initial USolids implementation phased out in 2018
  - entered production phase for VecGeom
  - adopted Apache 2.0 license





- continuation of USolids project started in AIDA
- extended scope to eventual work for GeantV with parallel navigation

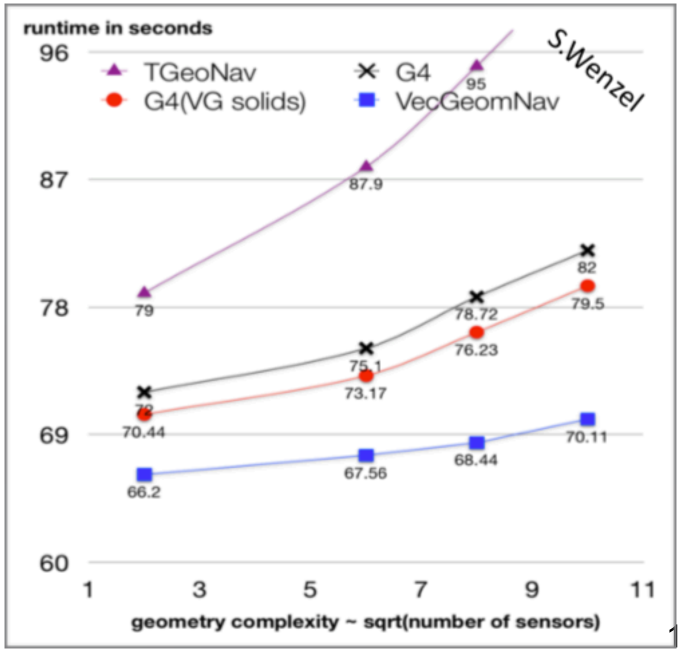


Internal algorithm vectorization
internal loop over lateral planes for distance calculation
Beneficial for current simulations

VecGeom - Latest Developments

G.Cosmo

- all relevant shapes implemented
  - can be used in Geant4 and ROOT
  - extended w/ generation of polyhedral meshes for visualization/debugging
- implemented Geant4 compatible navigation
  - using SIMD instructions
  - -> improved performance
- VecGeom used in production by CMS
  - planned for ATLAS
- future work:
  - use VecGeom on GPUs
  - continue work on G4 navigation



AIDA 2020

Advanced tracking

H.Grasland

Belle 2 profiling

- Belle 2 tracking uses genfit with a Geant4 geometry
  - Slow + little maintenance → Desire to move to Acts
  - Few people available → Desire for a **piecewise integration**
  - But what would be a promising candidate?
- I profiled the Belle 2 reco to answer this question
  - 40% of reco time spent in genfit + children
  - Large chain of **non-inlined** hot function calls...
  - ...leading to G4Navigator calls (33% of genfit time)
- Next: Improve hot path inlining + try Acts geometry.

Acts micro-benchmarks

- Acts benchmark infrastructure underwent major rework
  - Multiple timed runs → Output timings now have error bars
  - Robust stats → Filter out OS scheduler timing noise
  - Warmup time → Study steady state, not transients+steady
  - Optimization barriers → No “unfair” code simplification
  - Lambda-friendly API → Very easy to write new benchmarks
- **Usage example:**

```
const auto fixedPos = genPos();
const auto map_cached_result = Acts::Test::microBenchmark(
  [&] { return bFieldMap.getField(fixedPos); },
  iters_map
);
std::cout << map_cached_result << std::endl;
```

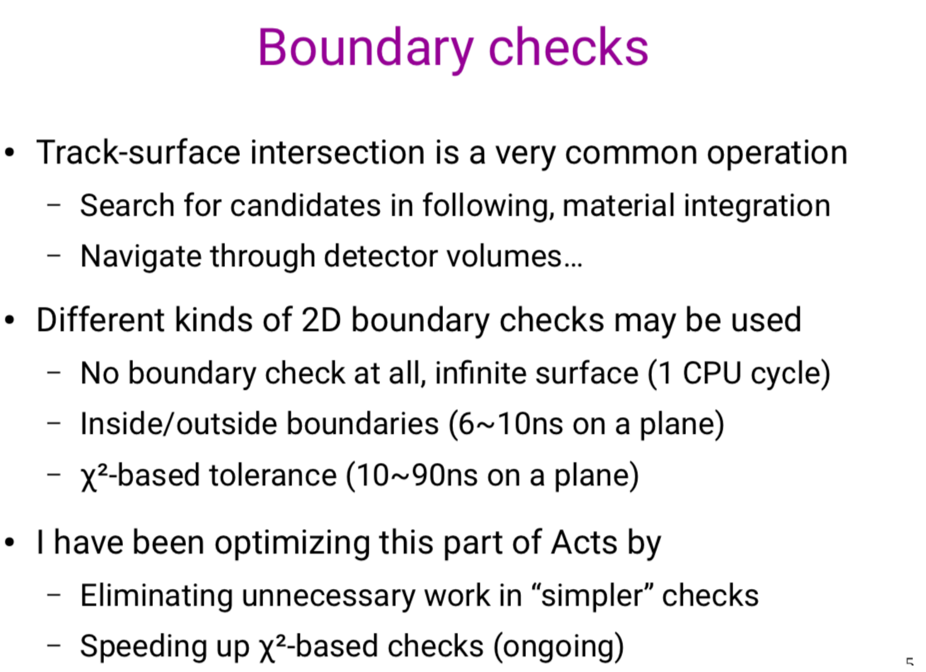
- **Output:**

```
20000 runs of 500 iteration(s), 591.7ms total,
29.5020+/-0.0363μs per run, 59.004+/-1.624ns per iteration
```

- task on advanced tracking tools has shifted focus to contribute to the community tracking toolkit ACTS
- focused on technically challenging topics

Advanced tracking

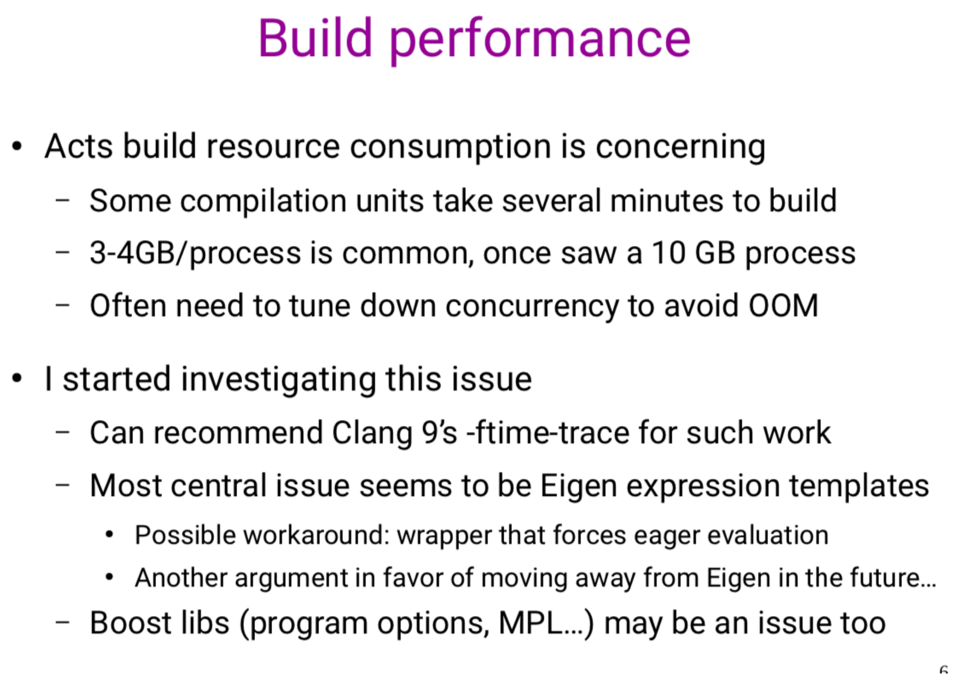
H.Grasland



Boundary checks

- Track-surface intersection is a very common operation
  - Search for candidates in following, material integration
  - Navigate through detector volumes…
- Different kinds of 2D boundary checks may be used
  - No boundary check at all, infinite surface (1 CPU cycle)
  - Inside/outside boundaries (6~10ns on a plane)
  - χ^2-based tolerance (10~90ns on a plane)
- I have been optimizing this part of Acts by
  - Eliminating unnecessary work in “simpler” checks
  - Speeding up χ^2-based checks (ongoing)

5



Build performance

- Acts build resource consumption is concerning
  - Some compilation units take several minutes to build
  - 3-4GB/process is common, once saw a 10 GB process
  - Often need to tune down concurrency to avoid OOM
- I started investigating this issue
  - Can recommend Clang 9’s -ftime-trace for such work
  - Most central issue seems to be Eigen expression templates
    - Possible workaround: wrapper that forces eager evaluation
    - Another argument in favor of moving away from Eigen in the future…
  - Boost libs (program options, MPL…) may be an issue too

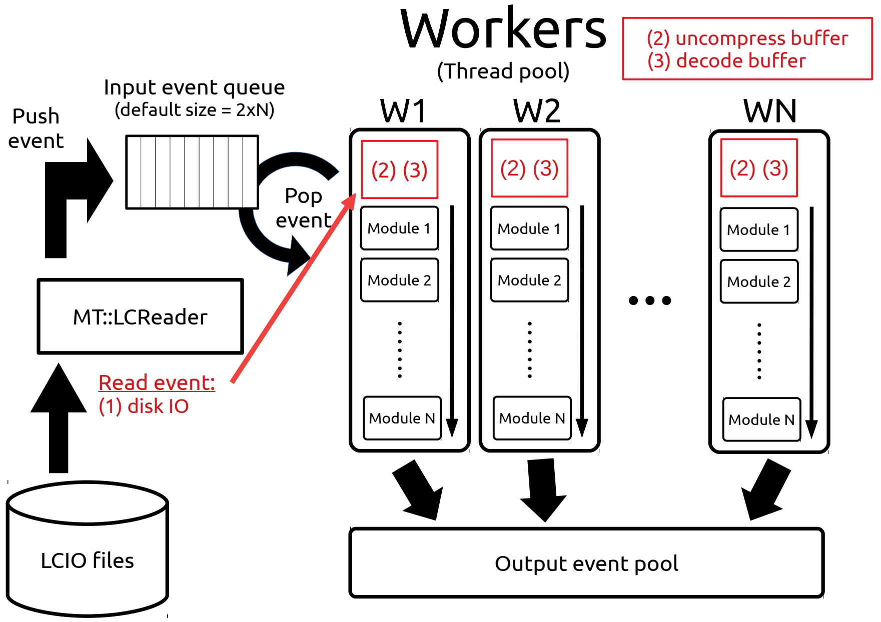
6

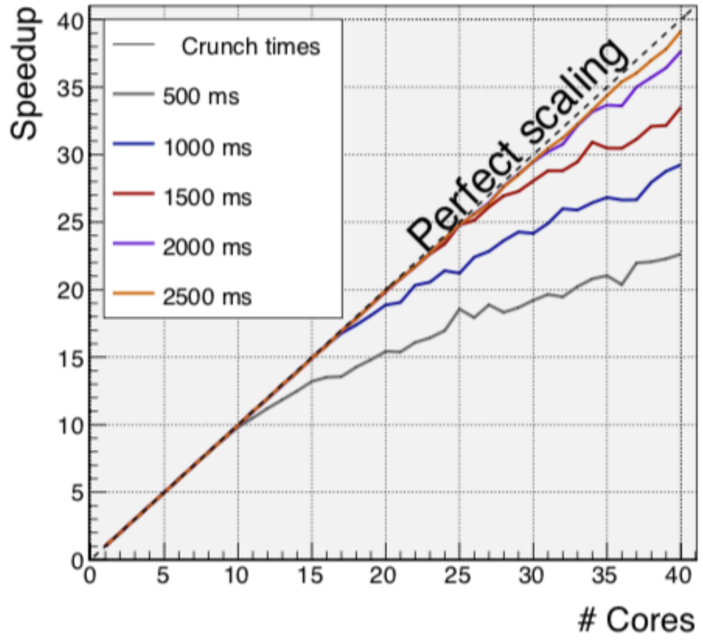
- task on advanced tracking tools has shifted focus to contribute to the community tracking toolkit ACTS
- focused on technically challenging topics

Framework Extensions

R.Ete

- **MarlinMT**: reimplementation of the Marlin framework – used throughout linear collider community - for parallel processing of events using multi-threading
  - includes re-implementation of LCIO I/O layer (SIO)
  - allows **lazy un-compression and event un-packing** in parallel worker threads

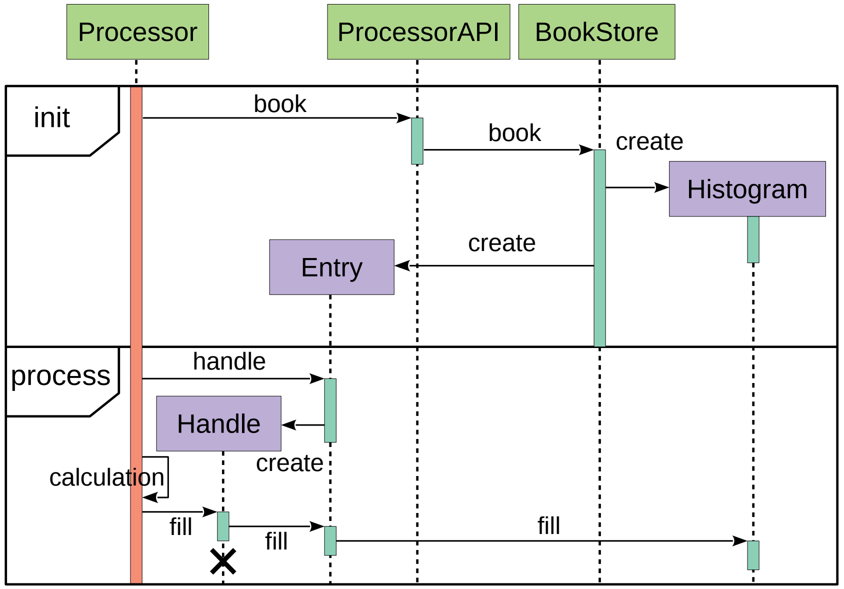


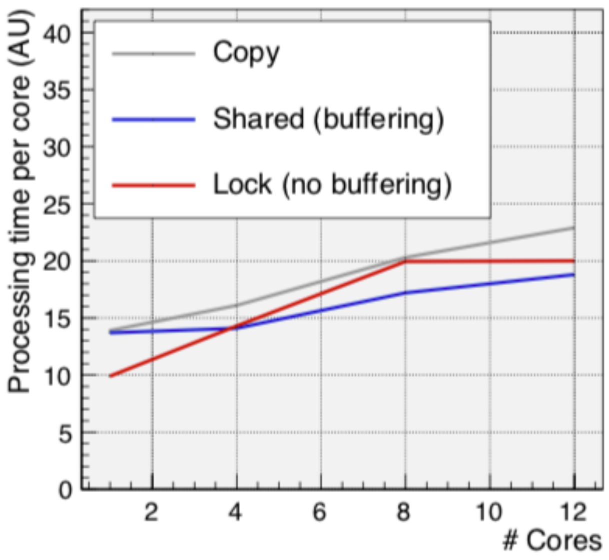


Framework Extensions

R.Ete

- **BookStore:** generic tool for the booking and filling of histograms in parallel processing
  - allows to transparently choose between **single-locking**, **bulk-filling** and **copy per thread**
  - based on new ROOT7 histogram library, built with ROOT 6
  - final histograms are converted to ROOT6 (backward compatibility of user macros)
- **applicable to any parallel HEP processing framework**





AIDA 2020

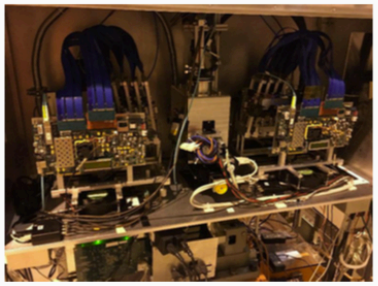
Alignment

S.Borghi

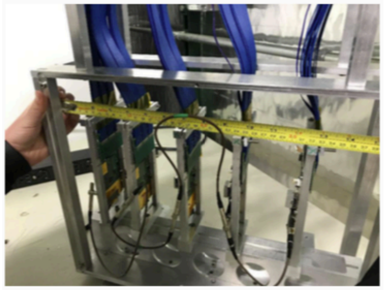
Alignment Toolkit

- The alignment toolkit is fully functional and can be used with DD4hep geometries to extract alignment conditions
- Used extensively in LHCb testbeams for upgrade:
  - Timepix telescope at CERN
    - Comprised of 8 timepix sensors
    - Sensors can be rotated around x/y to improve resolution/charge sharing
    - Remotely movable (Tx/Ty/Ry) DuT can be placed between the arms
  - Velopix telescope at Fermilab
    - Comprised of velopix sensors (+ DuT)
- Other users: LHC beamgas vertex group and MICE (Muon Ionising Cooling Experiment)

Timepix telescope



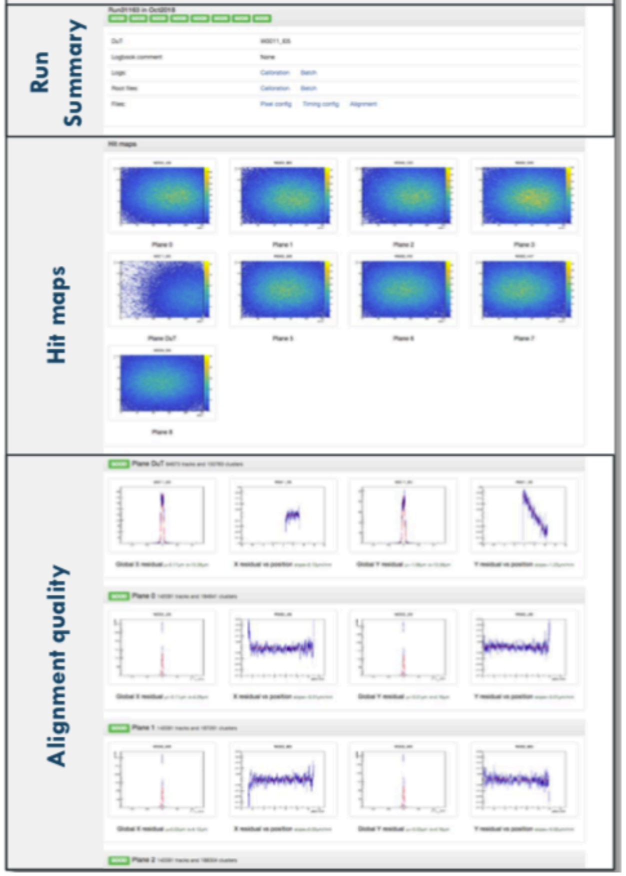
Velopix telescope



Application of alignment toolkit

- Extendently used in the LHCb testbeam with a timepix telescope.
- Thousands of datasets have been successfully aligned.
- Web application to monitoring the status and the quality of the alignment using Python's flask framework was developed.

Summary page of the alignment status per run

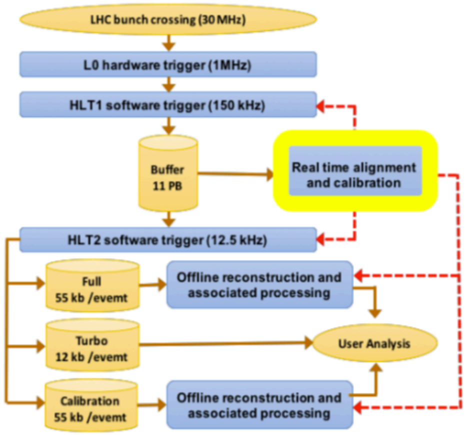


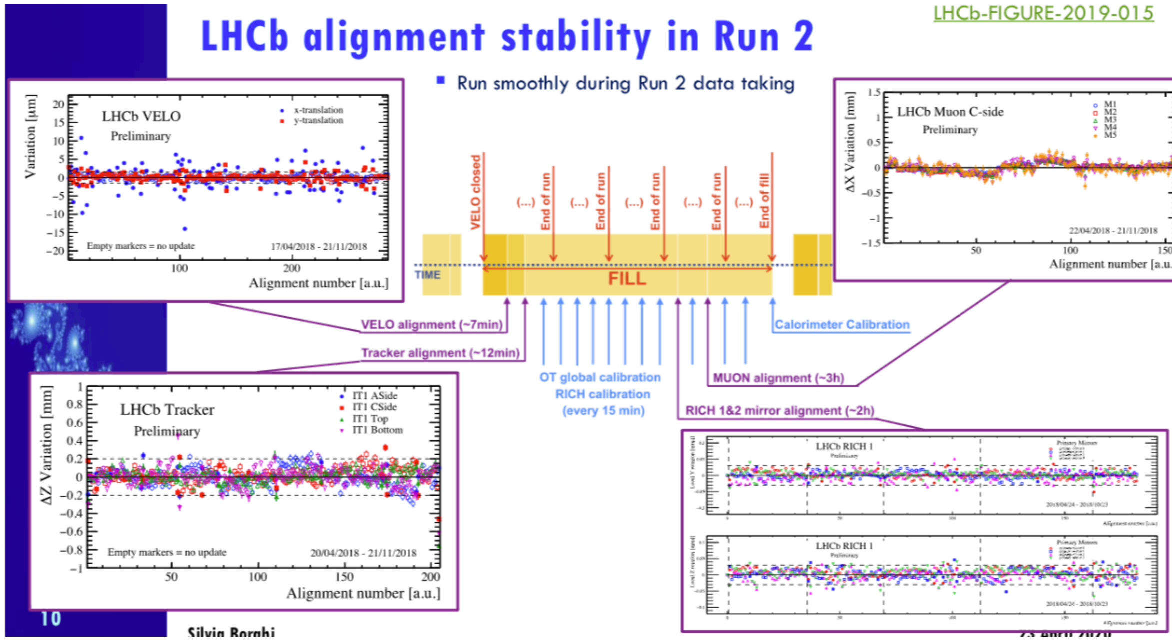
Alignment

S.Borghi

Real-time alignment at LHCb

- In Run2, a innovative real-time alignment procedure was developed
- Alignment of the full tracking system (VELO, UT, OT, IT) runs automatically in few minutes at the beginning of the fill.
  - Reconstruction is parallelized across ~1700 nodes of the online farm and the minimization is performed in a single node
  - Several thousands of elements are aligned.
  - Update of the constants if they are significantly different
- The full detector alignment is available in the second stage of the software trigger: one of the key elements to have the same online and offline performance


LHC bunch crossing (30 MHz)
L0 hardware trigger (1MHz)
HLT1 software trigger (150 kHz)
Buffer 11 PB
Real time alignment and calibration
HLT2 software trigger (12.5 kHz)
Full 55 kb /evemt
Offline reconstruction and associated processing
Turbo 12 kb /evemt
User Analysis
Calibration 55 kb /evemt
Offline reconstruction and associated processing



real time alignment procedure for LHCb considered pioneering work for HL-LHC era

AIDA 2020

EDM Toolkit

FG

Reminder: main PODIO features

- PODIO EDM toolkit based on the use of **PODs**
- originally developed in context of FCC
- application to *CLIC/ILC* (**LCIO**) planned from the start

- main features:
  - three implementation layers
  - well defined object ownership
  - relations between objects
  - C++ code generation with Python
  - Python binding/interface

Status at Annual Meeting 2019

Name	What	When
MS19	Design document for EDM Toolkit	M14
MS90	Application of EDM Toolkit to LC	M44
D3.4	Event Data Model Toolkit	M40

status

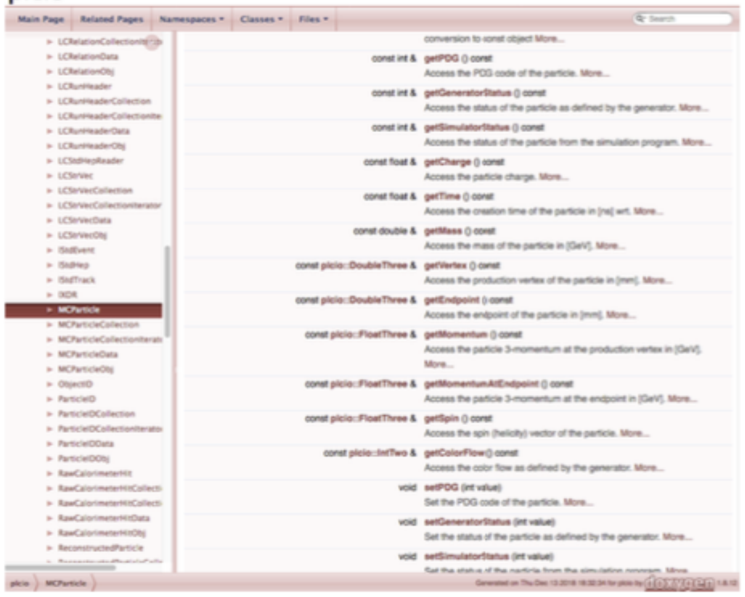
- all Milestones and Deliverables **reached on time**
- since then continued to improve PODIO

Developments since last Annual Meeting

- finalized pLCIO
  - re-implementation of LCIO in PODIO
- started new project: **EDM4hep**

- investigated implementation of **HDF5** I/O layer
- implemented POD-based binary I/O layer with **SIO**
  - benchmarking of SIO performance

plcio



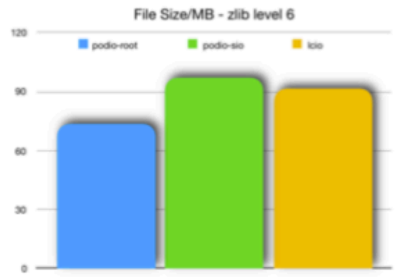
```
                               conversion to const object More...
                 const int &   getPDG () const
                               Access the PDG code of the particle. More...
                 const int &   getGeneratorStatus () const
                               Access the status of the particle as defined by the generator. More...
                 const int &   getSimulatorStatus () const
                               Access the status of the particle from the simulation program. More...
               const float &   getCharge () const
                               Access the particle charge. More...
               const float &   getTime () const
                               Access the creation time of the particle in [ns] wrt. More...
              const double &   getMass () const
                               Access the mass of the particle in [GeV]. More...
const plcio::DoubleThree &     getVertex () const
                               Access the production vertex of the particle in [mm]. More...
const plcio::DoubleThree &     getEndpoint () const
                               Access the endpoint of the particle in [mm]. More...
  const plcio::FloatThree &    getMomentum () const
                               Access the particle 3-momentum at the production vertex in [GeV].
                               More...
  const plcio::FloatThree &    getMomentumAtEndpoint () const
                               Access the particle 3-momentum at the endpoint in [GeV]. More...
  const plcio::FloatThree &    getSpin () const
                               Access the spin (helicity) vector of the particle. More...
     const plcio::IntTwo &     getColorFlow() const
                               Access the color flow as defined by the generator. More...
                      void     setPDG (int value)
                               Set the PDG code of the particle. More...
                      void     setGeneratorStatus (int value)
                               Set the status of the particle as defined by the generator. More...
                      void     setSimulatorStatus (int value)
```

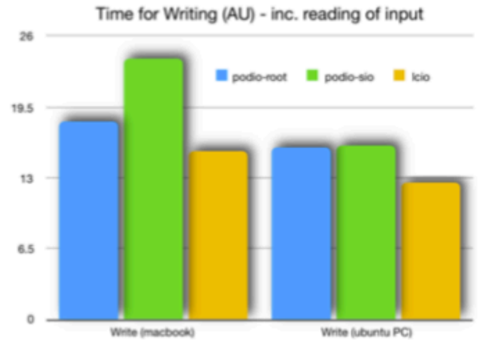
Generated on Thu Dec 13 2018 18:32:34 for plcio by doxygen 1.8.12

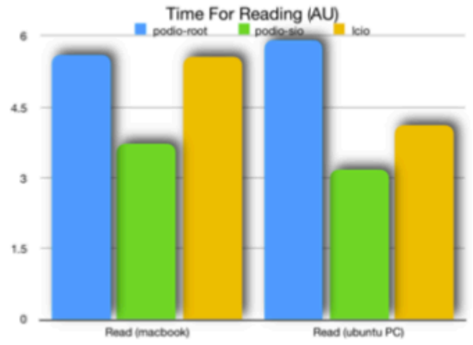
EDM Toolkit

FG

Benchmarking the ROOT vs SIO implementation







- ROOTI/O file size is ~**76%** of SIO file size
- ROOTI/O writing takes **75-99%** of SIO writing time [1]
- ROOTI/O reading takes **150-186%** of SIO reading time

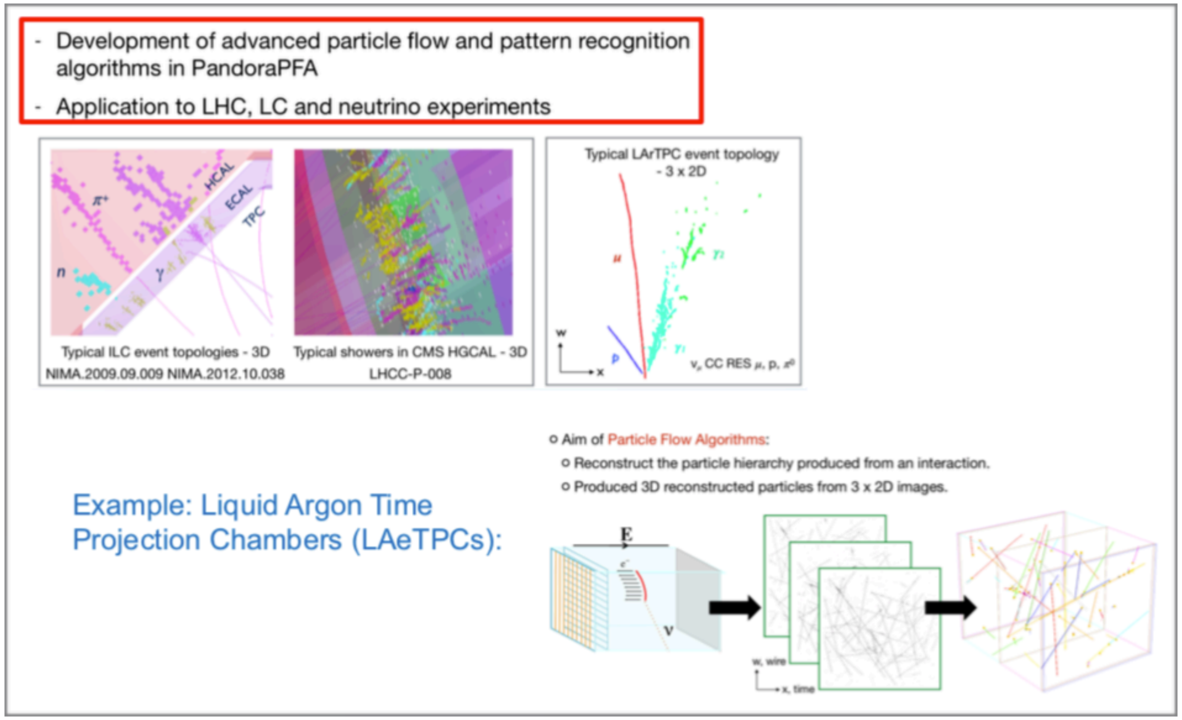
Next Steps and Plans

- continue the work on the **HDF5** implementation
- generalize the use of **different I/O implementations**
  - HDF5, SIO, potentially others ? . . .
  - need to iterate on and standardize the interface between *EventStore* and *Reader/Writer* implementations

- work planned in AIDAinnova successor project to AIDA2020

ParticleFlow Tools

- no report in last meeting as task coordinator has left the field and people have moved from UCam to other institutes
  - little activity in year 5 in context of AIDA2020
  - PandoraPFA well maintained and continued to be used in LC, neutrino and future colliders (AIDAinnova)



- Development of advanced particle flow and pattern recognition algorithms in PandoraPFA
- Application to LHC, LC and neutrino experiments

Typical ILC event topologies - 3D NIMA.2009.09.009 NIMA.2012.10.038

Typical showers in CMS HGCAL - 3D LHCC-P-008

Typical LArTPC event topology - 3 x 2D

Example: Liquid Argon Time Projection Chambers (LAeTPCs):

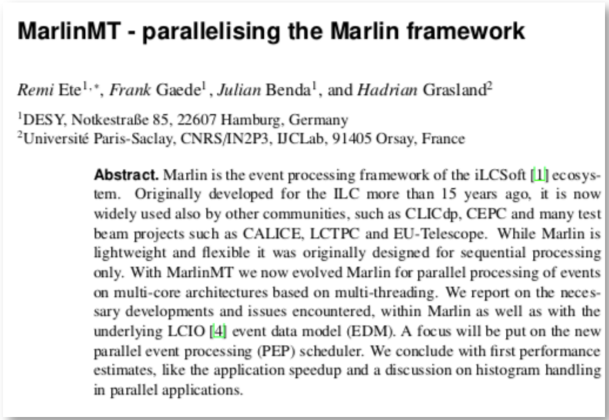
- Aim of Particle Flow Algorithms:
  - Reconstruct the particle hierarchy produced from an interaction.
  - Produced 3D reconstructed particles from 3 x 2D images.

PandoraPFA continues to be the prime framework for PFA developments for future colliders

Outlook AIDAinnova

- planned software WP for AIDAinnova:
  - some continuation of successful projects - some new activities
- **TurnKey** software stack
  - generic software stack for future colliders and HEP experiments
  - DD4hep, PODIO/EDM4hep, ACTS, Pandora, Gaudi,…
- **Simulation**
  - fast simulation with Machine Learning, integrated in Geant4
- **Track Reconstruction**
  - further develop ACTS and integrate in Key4hep, add (ML) pattern recognition and heterogenous computing
- **Particle Flow Reconstruction**
  - extend PFA algorithms to new calorimeter technologies and integrate in Key4hep; improve LAr detector algorithms

Publications

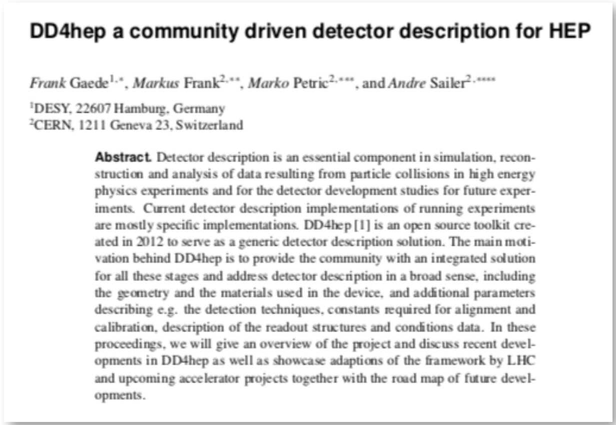


MarlinMT - parallelising the Marlin framework

Remi Ete[1,*], *Frank* Gaede[1], *Julian* Benda[1], and *Hadrian* Grasland[2]

[1]DESY, Notkestraße 85, 22607 Hamburg, Germany
[2]Université Paris-Saclay, CNRS/IN2P3, IJCLab, 91405 Orsay, France

Abstract. Marlin is the event processing framework of the iLCSoft [1] ecosystem. Originally developed for the ILC more than 15 years ago, it is now widely used also by other communities, such as CLICdp, CEPC and many test beam projects such as CALICE, LCTPC and EU-Telescope. While Marlin is lightweight and flexible it was originally designed for sequential processing only. With MarlinMT we now evolved Marlin for parallel processing of events on multi-core architectures based on multi-threading. We report on the necessary developments and issues encountered, within Marlin as well as with the underlying LCIO [4] event data model (EDM). A focus will be put on the new parallel event processing (PEP) scheduler. We conclude with first performance estimates, like the application speedup and a discussion on histogram handling in parallel applications.

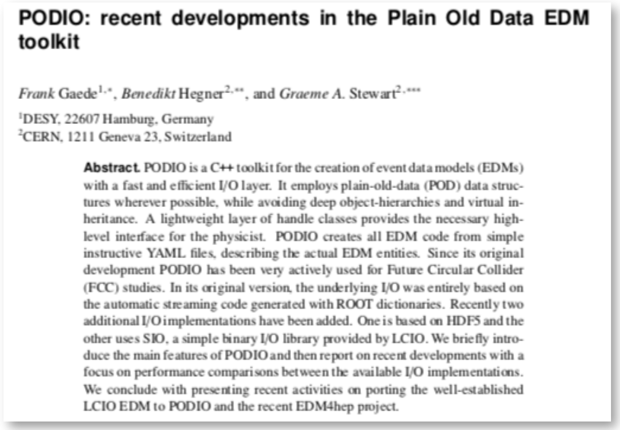


DD4hep a community driven detector description for HEP

Frank Gaede[1,*], *Markus* Frank[2,**], *Marko* Petric[2,***], and *Andre* Sailer[2,****]

[1]DESY, 22607 Hamburg, Germany
[2]CERN, 1211 Geneva 23, Switzerland

Abstract. Detector description is an essential component in simulation, reconstruction and analysis of data resulting from particle collisions in high energy physics experiments and for the detector development studies for future experiments. Current detector description implementations of running experiments are mostly specific implementations. DD4hep [1] is an open source toolkit created in 2012 to serve as a generic detector description solution. The main motivation behind DD4hep is to provide the community with an integrated solution for all these stages and address detector description in a broad sense, including the geometry and the materials used in the device, and additional parameters describing e.g. the detection techniques, constants required for alignment and calibration, description of the readout structures and conditions data. In these proceedings, we will give an overview of the project and discuss recent developments in DD4hep as well as showcase adaptions of the framework by LHC and upcoming accelerator projects together with the road map of future developments.



PODIO: recent developments in the Plain Old Data EDM toolkit

Frank Gaede[1,*], *Benedikt* Hegner[2,**], and *Graeme A.* Stewart[2,***]

[1]DESY, 22607 Hamburg, Germany
[2]CERN, 1211 Geneva 23, Switzerland

Abstract. PODIO is a C++ toolkit for the creation of event data models (EDMs) with a fast and efficient I/O layer. It employs plain-old-data (POD) data structures wherever possible, while avoiding deep object-hierarchies and virtual inheritance. A lightweight layer of handle classes provides the necessary high-level interface for the physicist. PODIO creates all EDM code from simple instructive YAML files, describing the actual EDM entities. Since its original development PODIO has been very actively used for Future Circular Collider (FCC) studies. In its original version, the underlying I/O was entirely based on the automatic streaming code generated with ROOT dictionaries. Recently two additional I/O implementations have been added. One is based on HDF5 and the other uses SIO, a simple binary I/O library provided by LCIO. We briefly introduce the main features of PODIO and then report on recent developments with a focus on performance comparisons between the available I/O implementations. We conclude with presenting recent activities on porting the well-established LCIO EDM to PODIO and the recent EDM4hep project.

- WP3 so far has **16 presentations**: 5 journals, 1 proceeding, 7 presentations, 3 notes
- a number of talks given at CHEP 2019 in Adelaide
- should be able to add more conference proceedings on AIDA2020 packages:
  - **DD4hep** detector description
  - **PODIO** EDM toolkit
  - **MarlinMT** processing framework
  - DD4hep in CMS
  - …

Conclusion

- the **advanced software work package** was overall very successful:
  - most of the developed packages are used already very broadly in the HEP community and have **achieved more than originally planned**
  - some tasks had to shift the focus in the course of the project and managed to provide **valuable** other **contributions**

- maybe the biggest achievement of the AIDA/AIDA2020 projects is a true **community building** among the software experts from the different fields of particle physics and the now generally accepted believe in **common software tools** in HEP

- **we would like to thank everyone involved for the dedicated work, the lively discussions and the good spirit of the last five years**

- very much **looking forward to AIDA-innova**

Witek and Frank