Track 6:

Collaborative software and maintainability

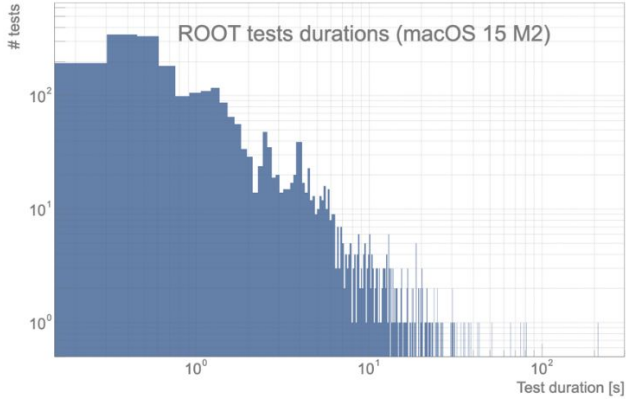
Wouter Deconinck
Matthew Feickert
Tobias Fitschen
Nathan Grieser

Collaborative software and maintainability

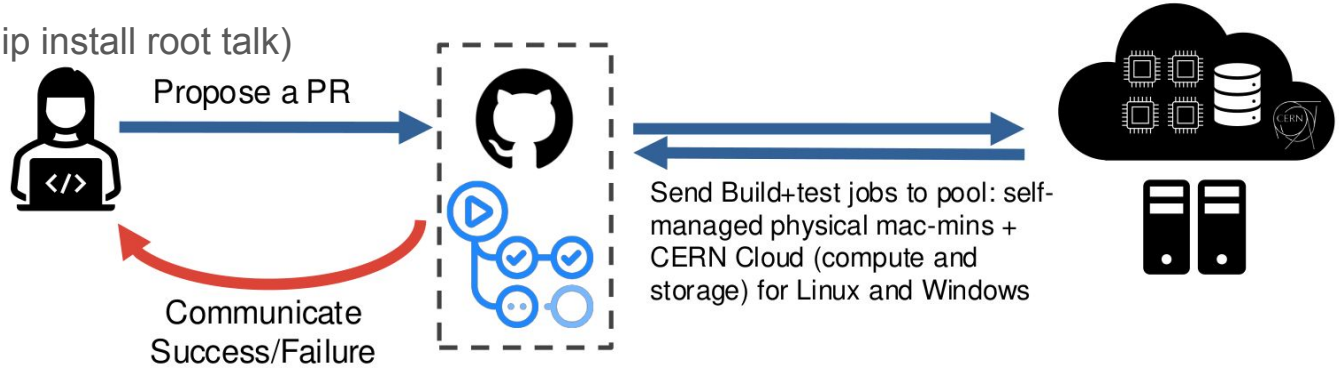
1. Containerization in the ATLAS Offline Code Management System
2. ATLAS ITk Production Database use and tools
3. XKIT for GridPP: (XRootD Kubernetes Integration Testing for GridPP)
4. dCache CI/CD migration to Kubernetes
5. Thoroughly testing and integrating hundreds of Pull Requests per month: ROOT's GitHub-based CI
6. Extending the Gaudi Software Framework outside of C++
7. Comparative efficiency of HEP codes across languages and architectures
8. pip install ROOT: experiences making a complex multi-language package accessible for Python users
9. AccGPT: A CERN Knowledge Retrieval Chatbot
10. LHCb Stripping Project: Continuing to Fully and Efficiently Utilize Legacy Data
11. Evolution of the ATLAS TRT Gas Gain Stabilization System Software
12. Enhancing software-hardware co-design for HEP by low-overhead profiling: AdaptivePerf
13. Thread-safe N-tuple Writing in Gaudi with TTree and Migration to RNTuple
14. A data Quality-Assurance framework for online and offline applications for the CBM experiment
15. Infrastructure for deployment and evaluation of LHCb trigger configurations
16. CERN OSPO's Catalogue Initiative: Amplifying Open Science for the CERN Community
17. Building the Key4hep Software Stack with Spack
18. Multi-package development at Fermilab with Spack
19. Collaborative software and maintainability for ePIC experiment at EIC
20. The ePIC Simulation Campaign Workflow on the Open Science Grid
21. The roles of the SCAB Nominations and Activities systems in the ATLAS-CERN speakers selection
22. Enhancing product management in the ATLAS Management Glance team

Thoroughly testing and integrating hundreds of Pull Requests per month: ROOT's new Cost-efficient and Feature Rich GitHub-based CI

Danilo Piparo



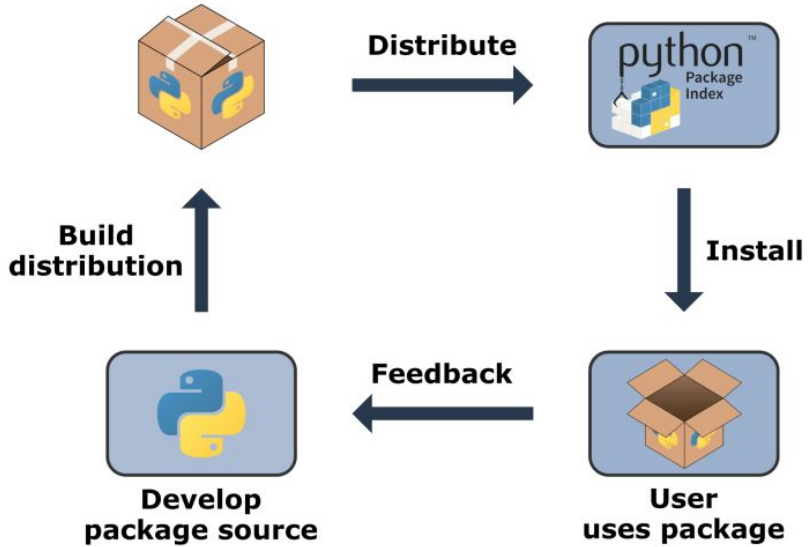
- Root
  - Open source on github (not gitlab since ROOT is not just CERN)
  - ~200 PRs per month (core devs + many external contributors)
  - 10k compilation targets, > 3k tests
  - Build+test for every PR + nightlies
- Moved from Jenkins CI to Github actions
  - Provides unified interface for more than just CI:
    - container upload to Harbor, doxygen documentation, jekyll build of ROOT website
  - Self-hosted github at CERN
    - Github runners not powerful enough for tests in reasonable time
- Future plans:
  - Python wheels (see pip install root talk)

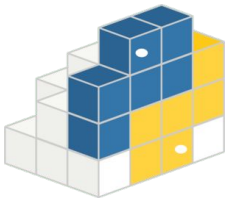


pip install ROOT: experiences making a complex multi-language package accessible for Python users

Vincenzo Eduardo Padulano

- Root has many existing distribution channels
  - Cvmfs, conda, system package managers (Linux/MacOS), docker images, binaries, Snap
- pip package was missing so far
  - Available now for Python 3.8+
  - Packaged as wheel
  - Only supported for x86_64 so far
  - Experimental (option -i + URL needed) so far
  - Provides root executable in terminal too





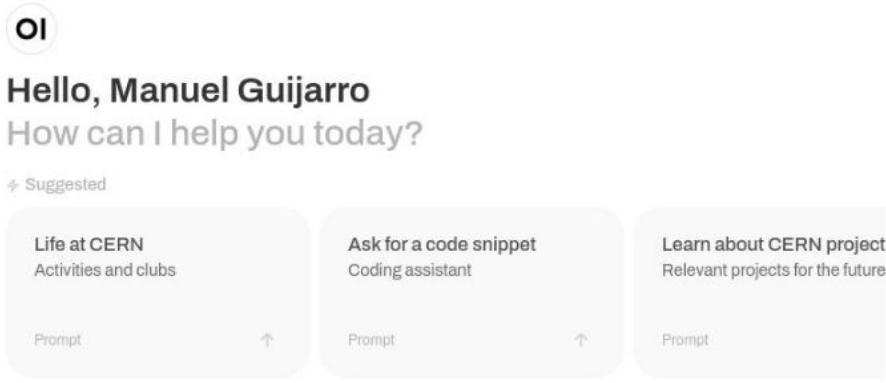
```
pip install ROOT -i https://root-experimental-python-wheels.web.cern.ch
```

AccGPT: A CERN Knowledge Retrieval Chatbot

Juan Manuel Guijarro

- Growing demand for LLM at CERN for various applications
- RAG on open-weight Meta LLama-3.1 (8B parameters)
  - RAG: Retrieval-Augmented Generation
  - Commercial+closed GPT4o did not show significant advantage
- Inputs
  - Internal CERN web pages (HTML + pdf)
  - Confluence spaces
  - CERN Document Server (CDS)
- WebUI and API (vLLM based)
- Future:
  - In-house code assistant?
  - Live note-taking?

Try out the chatbot



Enhancing software-hardware co-design for HEP by low-overhead profiling of single- and multi-threaded programs on diverse architectures with AdaptivePerf

Maksymilian Graczyk

- CERN project part of SYCLOP programme
  - Hardware acceleration with open standards using SYCL and RISC-V
  - Benchmarking, profiling, and integration testing of SYCLOPS use (ROOT, cling,...)
- AdaptivePerf
  - Open-source and language-agnostic code profiler build on top of perf
  - Sampling both CPU and non-CPU activity
  - Interactive flame graphs and charts in a web browser
  - Available on github
- Future plans
  - Support heterogeneous and custom hardware
  - Including CPU, GPU, FPGA, ASIC
  - Unified profiling system for CERN and HEP

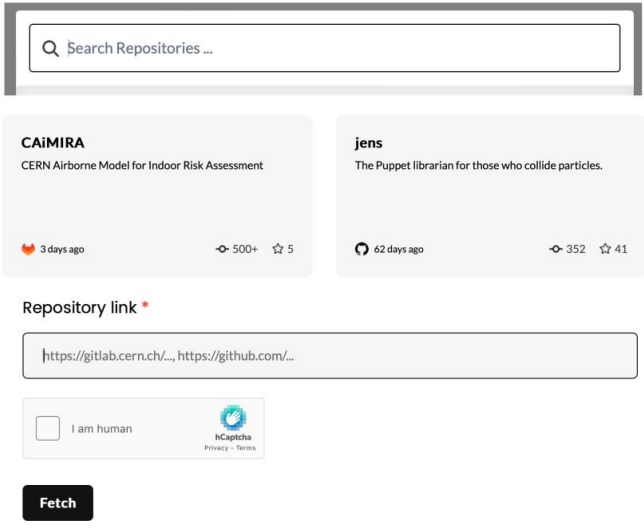
CERN OSPO's Catalogue Initiative: Amplifying Open Science for the CERN Community

Victoria Sigcha et al.

- OSPO Mandate
  - Provide a catalogue of CERN's open source that's accessible
- The OSPO (Open source programme office) Catalogue
  - Made in CERN and for CERN
  - A place to find similar projects and build collaboration
  - Accessible on any device (phone, laptop, …)
- How to get added?
  - Submission form on catalogue web page itself

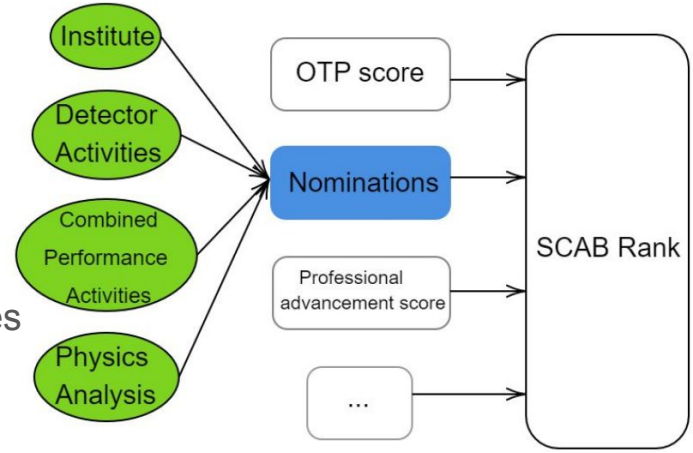
OBJECTIVE

- search, filter, tags
- essential info.
- submission forms
- content managing

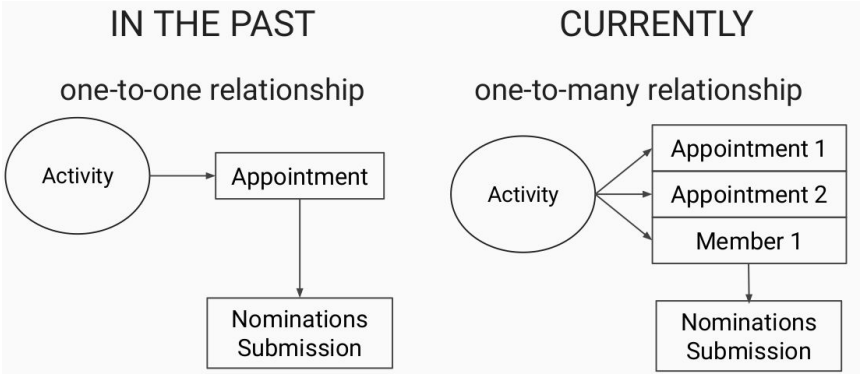


The roles of the SCAB Nominations and Activities systems in the ATLAS-CERN speakers selection

Carolina Rodrigues



- Glance
  - > 6k active members, ~250 institutes from 40 countries
  - Streamlines management of the experiment
  - Centralizes info about people, activities, publications, and roles
- Speakers Committee Advisory Board (SCAB)
  - Interface for members, papers, and speaker selection
  - Nominations system guarantees that all contributing members participate in selection for talks
  - SCABBER algorithm calculates score based on multiple nominations
- Adapted Domain Driven Design (DDS)
  - Maintainability, testability, ease of integration
  - Divides the code into bounded contexts
  - Structuring the software around its core domain



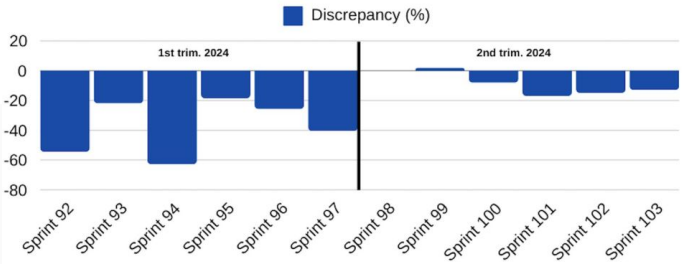
ATLAS EXPERIMENT

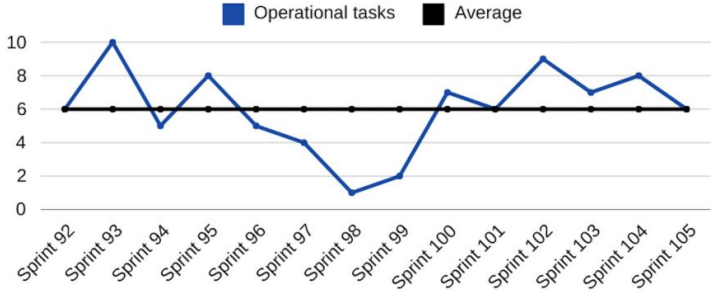
Glance

Enhancing product management in the ATLAS Management Glance team

Gabriela Lemos Lucidi Pinhao

- ATLAS Management Glance Team (AMGT)
  - Subset of Glance project
- AMGT introduced Product Owner (PO)
  - Role identified by SCRUM methodology
  - Serves as the main point of contact between devs and stakeholders
  - Now centralised in AMGT
- Backlog and roadmap management
  - Introduction of PO lead to clean-up of outdated tasks
  - Regular tracking of operational tasks favours system health and maintainability
  - Planning start/end date for ner-term roadmap items
  - PO centralizes stakeholder interactions, shielding devs from interference (mails, meetings)
  - Removed deprecated documentation, introduced glossary and overview





ATLAS EXPERIMENT Glance

Evolution of the ATLAS TRT Gas Gain Stabilization System Software

Bartosz Mindur

- ATLAS TRT GGSS
  - Compact system on single PC in LHC USA15 cavern
- Requirements
  - Continuous operation (every time ATLAS is powered on, not just during stable beam)
  - Failure resistant (hot-swappable hardware spares)
  - Easily monitored by non-experts
- Software evolved over 20 years of operation
  - Evolved from a few independent Windows console programs to > dozen git repos with submodules
- Good design pays off
  - Core functionality was never changed
  - GGSS software never failed during LHC operation

GGSS box with straws

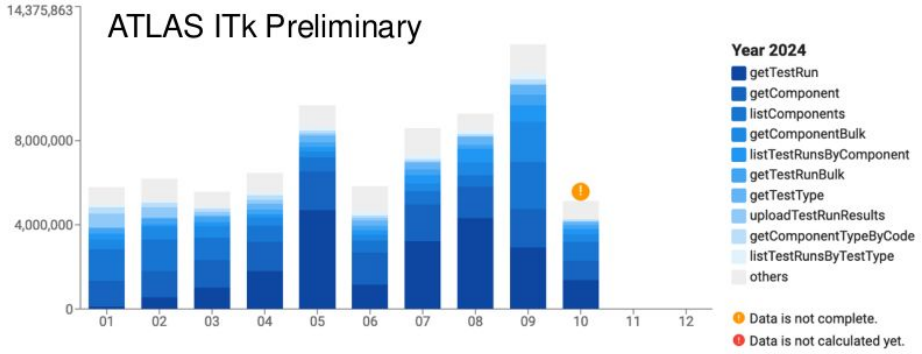
GGSS simplified algorithm

ATLAS EXPERIMENT

ATLAS ITk Production Database use and tools

Monika Wielers

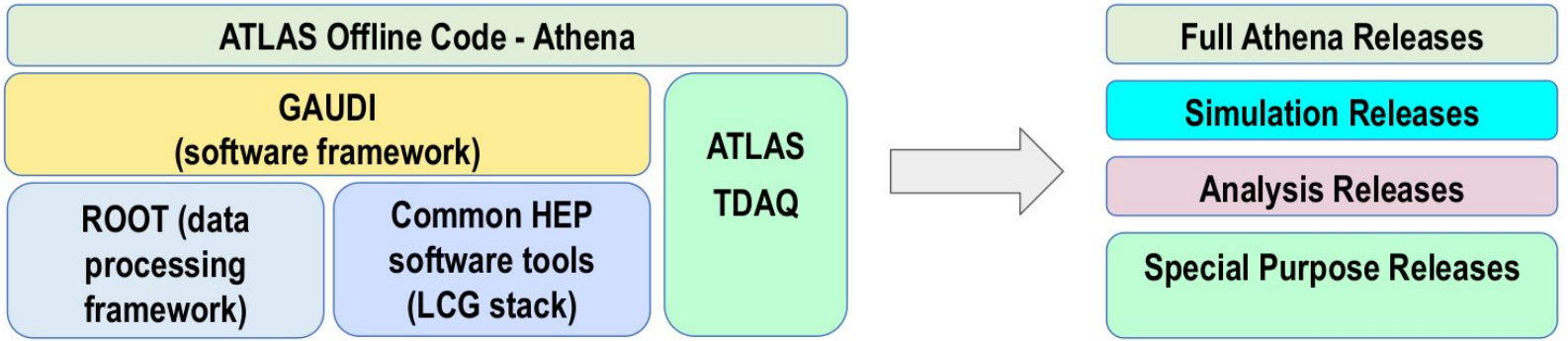
- ITk: new ATLAS all-silicon inner tracking detector
- Need database to keep track of full production project
  - 100 institutes from 22 countries + industry partners
  - ~10^6 parts to be produced
- Backend based on mongoDB + uuApp Framework developed by vendor
  - Front-end developed by ITk community
  - Up to 50 users can use DB simultaneously
  - Mostly open source, cloud-based (uuCloud on MS Azure)
  - API allows batch-upload in csv/excel format for labs
  - On-demand + scheduled reports for data consistency, monitoring rates, quality, yields
  - In use for 4 years now, includes 10^6 components + 10*10*6 test results
- Lessons learned
  - Important to tailor tools to diverse community
    - Technicians, engineers, physicists…



Containerization in the ATLAS Offline Code Management System

Chris Lee

- ATLAS: > 5M lines of C++/Python code
  - Requires > 200 external packages
  - 500 developers, 1k MRs annually
- ATLAS started adopting containers shortly after Docker release (2013)
  - Fully integrated into analysis workflows in 2022/23
  - Container storage on CERN's Harbor Container Repository (initially: Docker Hub)
  - Stable release containers are delivered to the grid via cvmfs
- Major Application: Legacy branches built in CentOS7 containers
  - Running on EL9 machines after 2024 CentOS7 phase-out
  - Analysis preservation

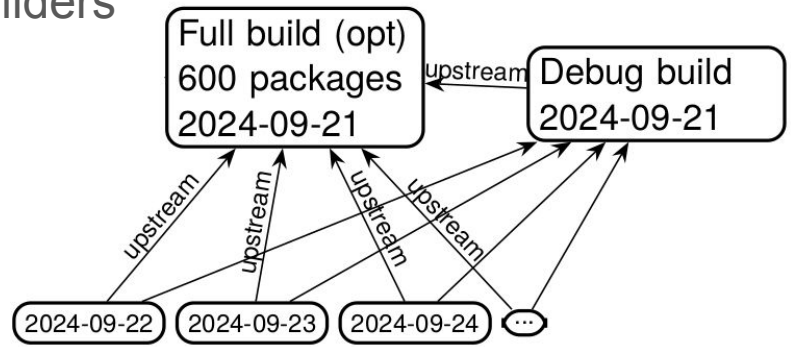


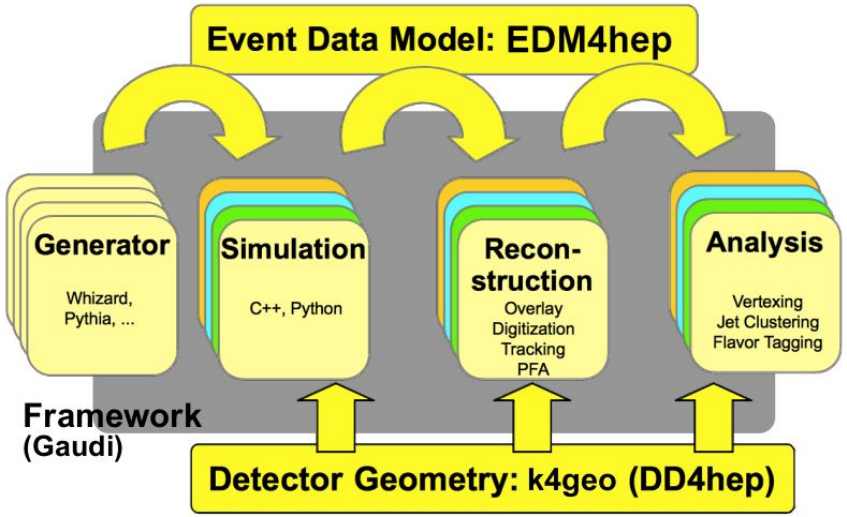
ATLAS EXPERIMENT

Building the Key4hep Software Stack with Spack

Juan Miguel Carceller

- Key4hep: Turnkey software for future colliders
  - EDM4hep: Event Data Model
  - k4FWCore: interface to Gaudi
  - k4geo: detector models
  - And many more…
- Spack
  - Multi-platform package manager
  - Supports multiple versions of packages, configurations, compilers
  - Configuration in yaml files
- Key4hep build
  - Around 600 packages + dependencies
  - Built with spack, deployed to cvmfs
  - Key4hep-stack package has all other packages as dependencies, pulls them when invoked
  - Debug builds upstream optimised builds





Spack

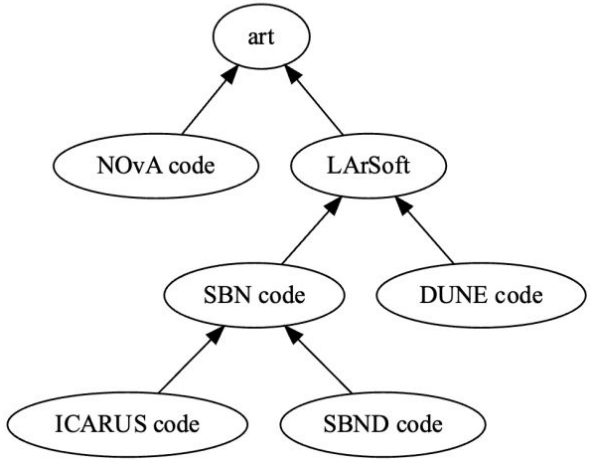
Multi-package development at Fermilab with Spack

Kyle Knoepfel

- Fermilab switched from in-house package manager to Spack
  - Engage and influence broader community outside of Fermilab
  - Benefit from input from large user base
- Created Spack extension: spack mpd
  - For orchestration builds of multiple CMake projects
  - Requires minimal Spack expertise
  - Easy to setup, switch between sessions, list available projects
  - Allows to merge many builds seamlessly
  - Requires good dependency management
  - Ready for beta-testing

https://github.com/FNALssi/spack-mpd

Simplified dependency graph



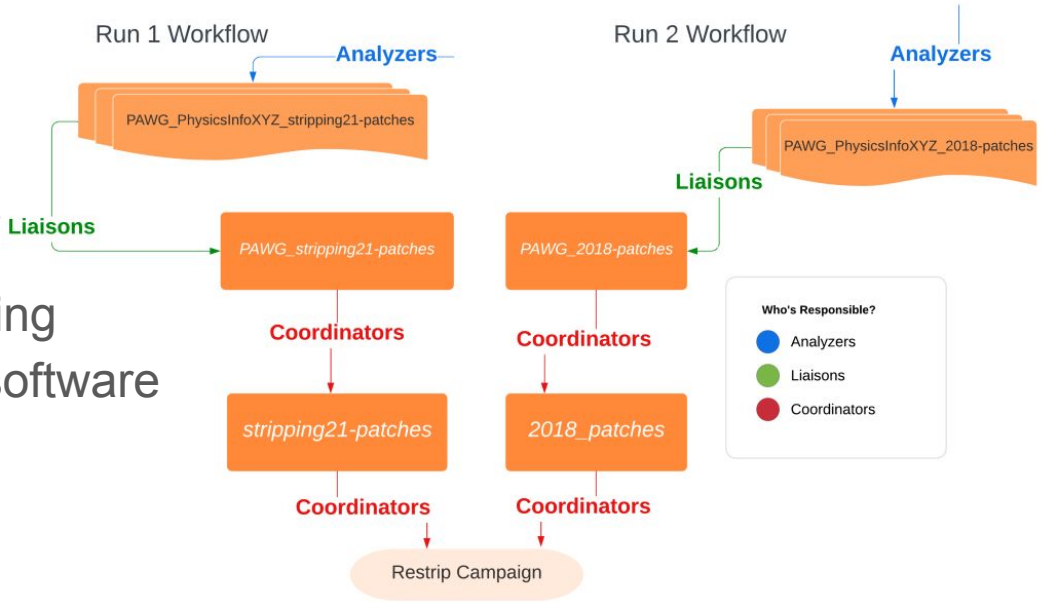
Spack

```
$ spack mpd status
==> Selected project: my-art-devel
    Environment status: active
```

LHCb Stripping Project: Continuing to Fully and Efficiently Utilize Legacy Data

Nathan Grieser

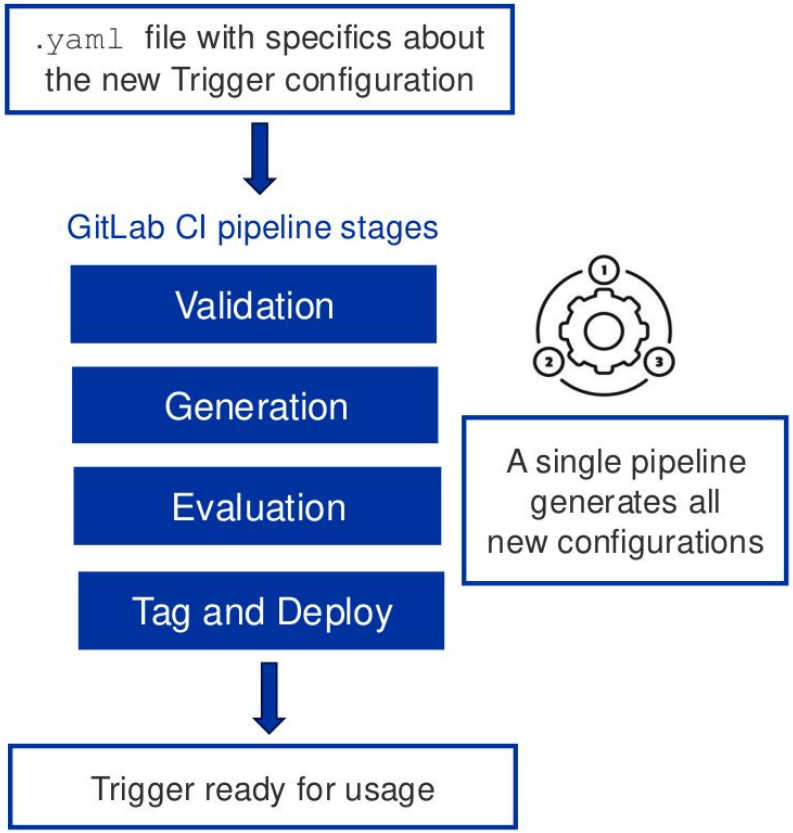
- Legacy software stack for LHCb Run1/2 is maintained
  - Allows to do analysis with new methods+ideas on existing data
- Liaisons from Physics WG responsible for WG-specific parts
  - Removes workload+pressure from coordinators
  - Analysts required to do the bookkeeping
- In depth training to adding roles, technical crash courses
- Tight production windows
  - Legacy analyses are allocated only limited resources
  - Deadlines for analysis steps are set and are strictly enforced
- Close collaboration with computing and operations/core software



Infrastructure for deployment and evaluation of LHCb trigger configurations

Luke Grazette

- Old procedure:
  - Manual generation of the trigger configurations via list of detailed instructions
- New procedure:
  - Infrastructure for generating and validating configs using gitlab CI pipelines
  - Automatically compares expectation to performance evaluation of the resultant trigger
  - Ensures consistency, reproducibility
  - Trigger operators have more ability to focus on the “higher level” information



New Workflow for creating a Trigger configuration

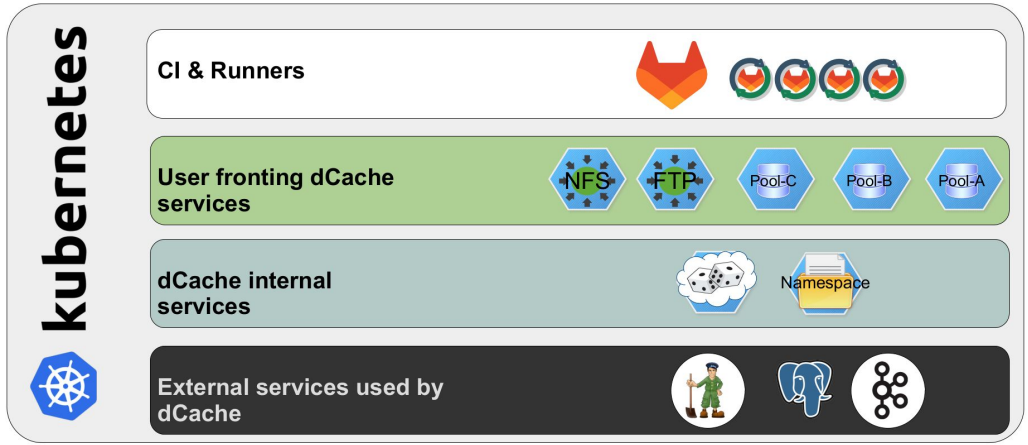
tck/0x1000109b.yaml 0 → 100644 +11 −0 Viewed

```
+ TCK: 0x1000108b
+ workflow: "commit-based"
+ parameters:
+     process: "Hlt1"
+     type: "odin_lumi"
+     label: "RTA/2024.09.29.2 with odin_lumi for emittance scan"
+     stack: "RTA/2024.09.29.2"
+     settings: "odin_lumi"
+     commits:
+       Allen: "255b6528f6ce53a0ffe7fa578beffc00e25b40bd"
+     comparison: 0x10001080
```

dCache CI/CD migration to Kubernetes

Tigran Mkrtchyan

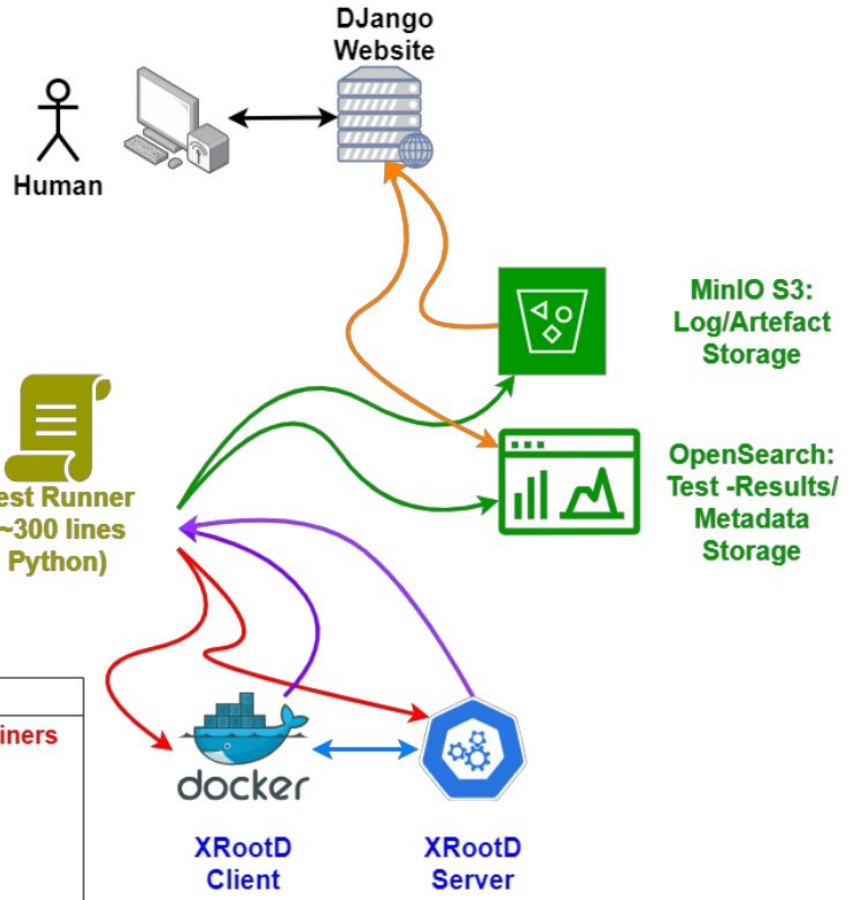
- dCache
  - Distributed storage system
  - For large amount of data, distributed among many heterogeneous server nodes
- Testing environment
  - Gitlab server provided by DESY-IT
  - Helm chart
  - CI pipeline
  - Containers
- Possible improvements
  - Kernel-specific tests have to be run manually
  - Yaml in Helm charts become verbose quickly



XKIT for GridPP: (XRootD Kubernetes Integration Testing for GridPP)

Wenlong Yuan

- XRootD:
  - Used by 5 large UK sites (and others)
  - All use different configurations + versions
- Containerisation
  - Using rpm build recipe to containerise XRootD
  - Hosted on Docker Hub
  - Service management: Kubernetes
- Supports monitoring of:
  - protoDUNE DAQ + DUNE-DM
  - GridPP-FTS
  - PPE for UoE labs clean-room certification

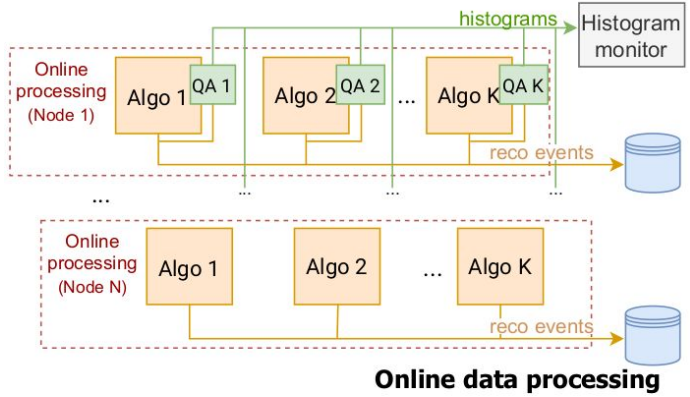


Testing Strategy
1) Deploy Configuration and Launch Containers
2) Wait for tests to run
3) Collect container artefacts
4) Store Test Results/Logs
4) Display Results to User

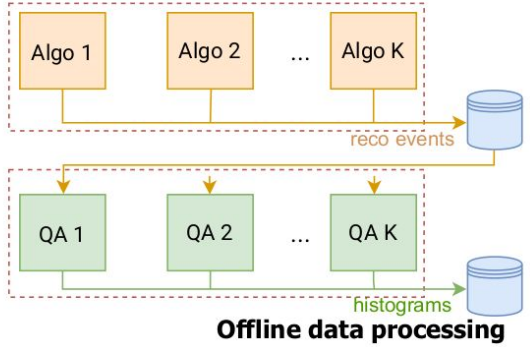
A data Quality-Assurance framework for online and offline applications for the CBM experiment

Sergei Zharko

- CBM@Fair
  - Fixed target experiment to study QCD phase diagram at large mu_B
  - Trigger reduces data rates from 1 TB/s to 10GB/s
  - Full online event reconstruction
- Separate QA frameworks for online/offline
  - Lightweight, fast online framework
  - With custom histograms classes based on boost::histogram (no ROOT dependencies)
- Recently harmonised online and offline QA
  - Failed checks are collected by the QA-manager and trigger CI-tests
  - Some properties can be ignored by operator, end up in final check-list status



Online data processing

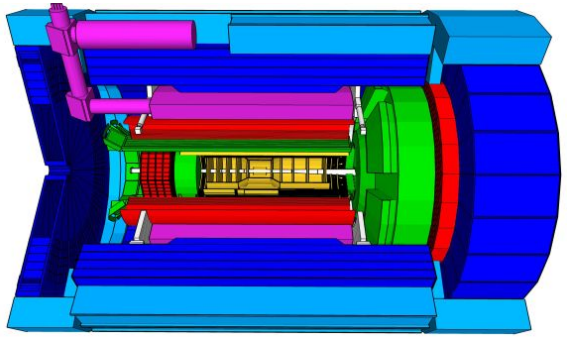


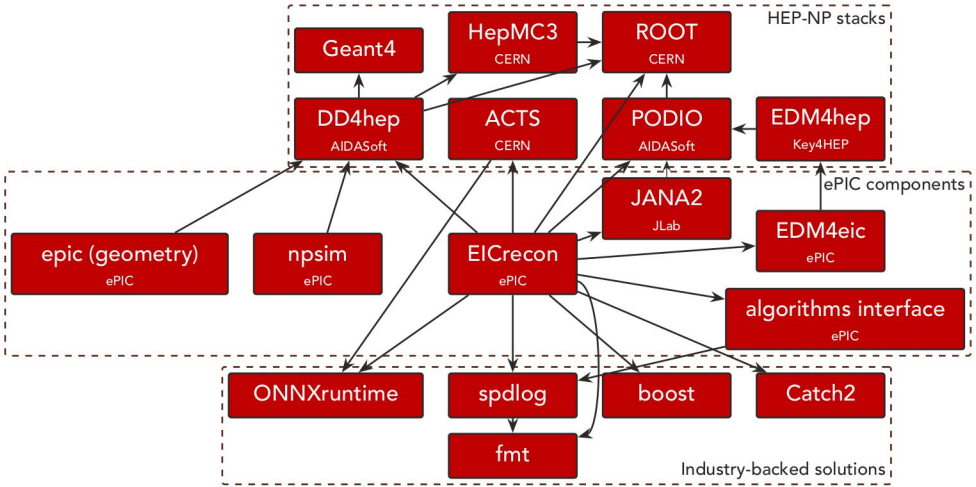
Offline data processing

Collaborative software and maintainability for ePIC experiment at EIC

Dmitry Kalinkin

- ePIC detector at EIC
  - Asymmetric, hermetic central detector
- EIC Software
  - Statement of principles
  - Agile development
  - Production-ready software stack
  - User-centered design
- Deployment and workflow
  - Publicly developed workflow
  - Github actions + gitlab CI
  - Snakemake
  - Artifacts are publicly accessible
  - Formatting, static & dynamic analysis
- ePIC Technical design report
  - Plots development conforming to EIC
  - Pre-TDR draft ready for review

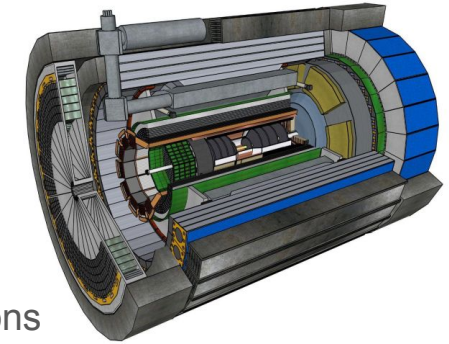




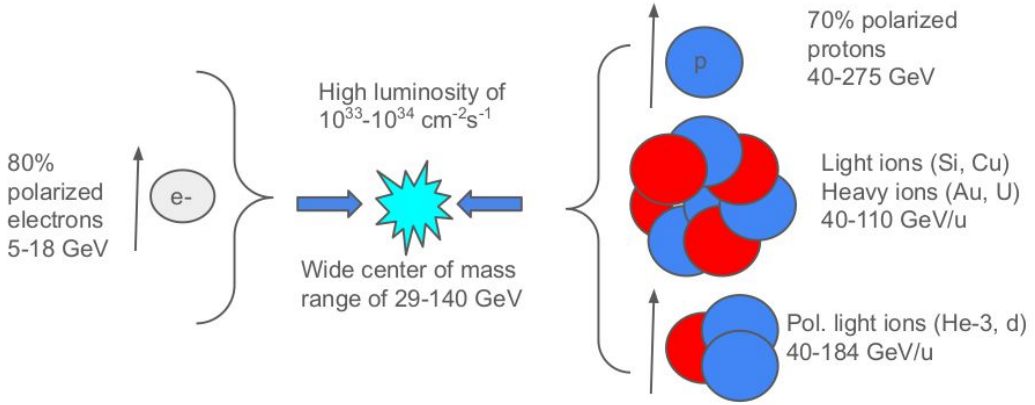
Modular Simulation, Reconstruction and Analysis Software Stack powered by the NP-HEP community tools

The ePIC Simulation Campaign Workflow on the Open Science Grid

Sakib Rahman



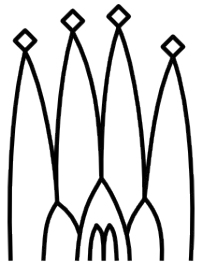
- Open Science Grid
  - distributed High Throughput Computing (HTC) for US HEP community
  - Can run 10k jobs simultaneously
  - HTCondor submit template
  - Excellent support system: Weekly meetings with OSG experts
  - CILogon provides access management from diverse institutional affiliations
- ePIC simulation campaigns
  - Consumed ~2k core-years on OSG
  - Continuous deployment of software for detector and physics studies
  - Regular update of production outputs
  - Timely validation and quality control

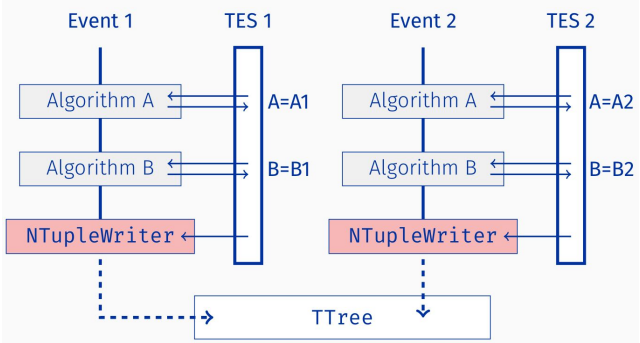


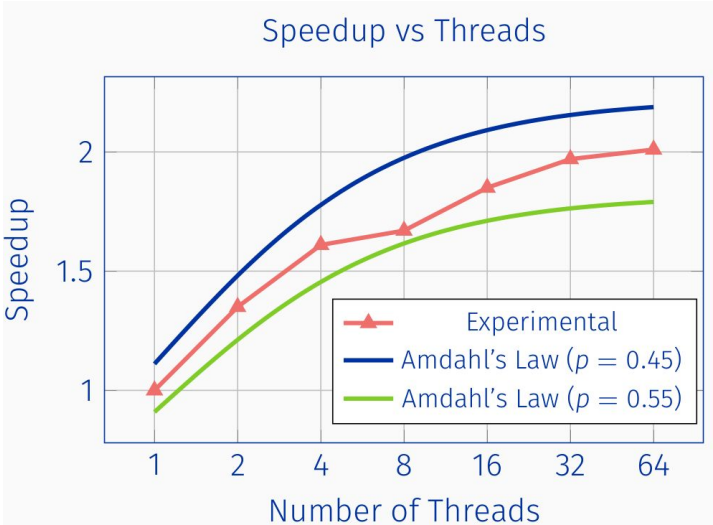
Thread-safe N-tuple Writing in Gaudi with TTree and Migration to RNTuple

Silia Taider

- Gaudi
  - C++ Framework to build data processing applications for HEP (ATLAS, LHCb, FCC…)
  - Multi-threading introduced in 2013, but parts remain non-thread safe
  - Uses TTree for nTuple writing which is inherently not thread-safe
  - RNTuple: Evolution of TTree, inherently columnar and thread safe
- Problem: NtupleSvc (sequential)
  - Single shared buffer that gets overwritten
  - May introduce race condition
- Solution: NtupleWriter (parallel, thread safe)
  - Allocate one buffer per event
  - Being tested in LHCb's analysis framework DaVinci



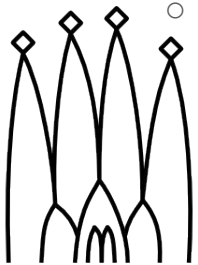


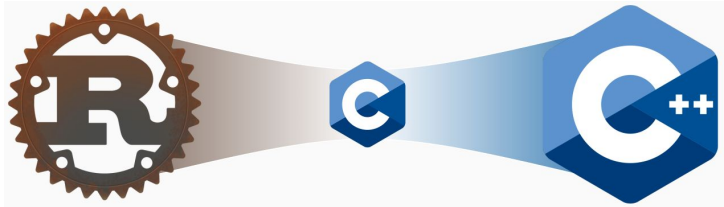


Extending the Gaudi Software Framework outside of C++

Marco Clemencic

- Gaudi:
  - C++ Framework to build data processing applications for HEP (ATLAS, LHCb, FCC…)
- Goal: Integrate safe Rust code into Gaudi, rather than rewriting everything
  - In C++ memory safety is an afterthought
  - Rust integrated with cxx + Corrosion
    - Memory safety as priority
    - Interoperable with C++
    - cxx: provides safe mechanism to call Rust code in C++ and vice versa
    - Corrosion: Tool for integrating Rust into existing CMake project
- Lessons learned
  - Extending a large C++ project like Gaudi with Rust is possible
  - Some (small) overhead is unavoidable (converting of container types etc)
  - Bindings and wrappers have to go through C APIs
    - Cannot use generics or templates (C++ features)

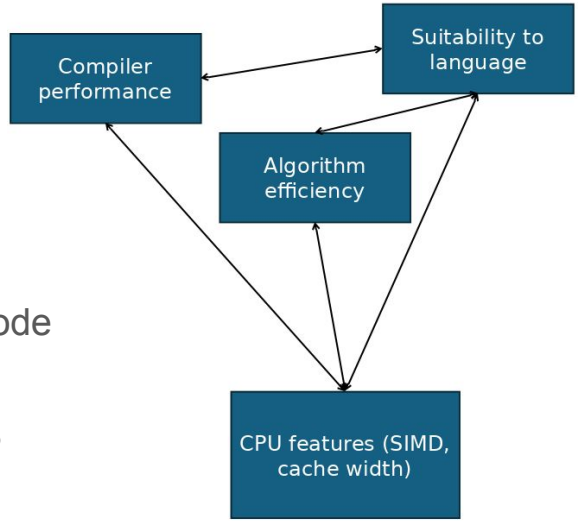


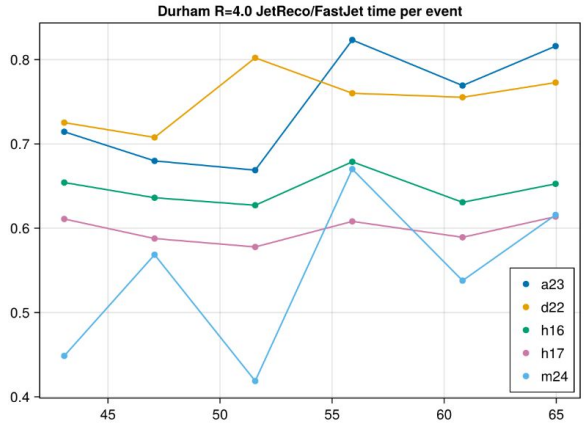


Comparative efficiency of HEP codes across languages and architectures

Samuel Cadellin Skipsey

- Julia
  - Performance close to C/C++/Fortran thanks to JIT
  - Integrated package management
  - LLVM-based like C/C++/Fortran
  - (Infered) typing: Rich expressive type system
  - @code_native macro yields in-place inspection of generated code
  - See also plenary talk Graeme Stewart
- Performed benchmarks on various CPU architectures
  - C++ FastJet vs Julia implementation JetReco
  - JetReco significantly faster in many test cases/configurations





HEP Software Foundation

Collaborative software and maintainability

Many thanks to the **contributors** and **participants** for the great discussions!

And to the **CHEP organisers** for providing a very productive environment!