Experience of operation the organized grid data analysis using Hyperloop train system

Vladimir Kovalenko (Saint Petersburg State University)

The author acknowledge Saint-Petersburg State University for a research project 103821868

Heavy-ion experiments

- NA61/Shine at SPS (CERN)
- ALICE at LHC (CERN)
- MPD at NICA (JINR, Dubna)

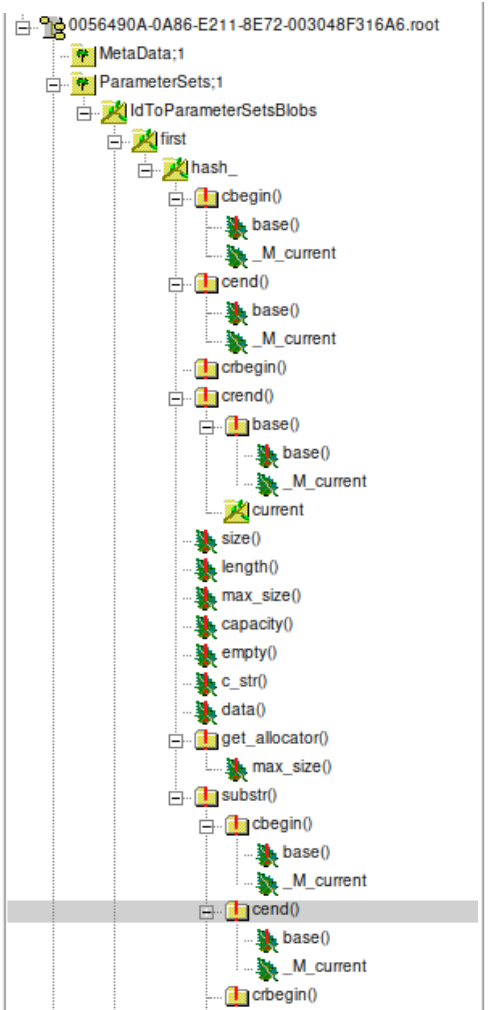
Data storage as ROOT objects

- Array, lists, trees of complicated objects (classes of events, tracks etc)
- Drawbacks: keep dedicated version of analysis software corresponding to the given data format
- Problems – software obsolescence, compatibility issues.
- Example: CERN open data (for ex. ALICE, CMS).
  - Data format is too complex
  - You can use only old software provided in virtual machines with obsolete versions of compiler and OS
  - Software will not build under modern OS

Array of structures (AoS)

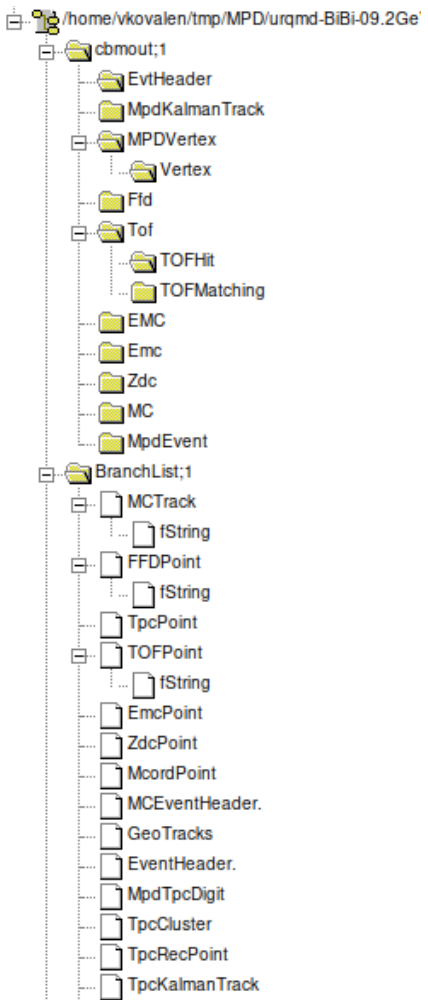
Structure of arrays (SoA):

CMS open data file

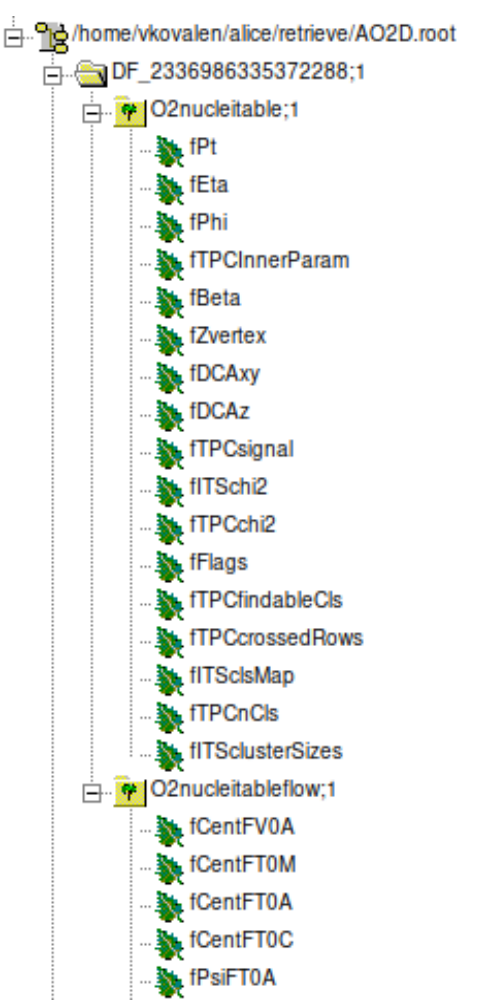


ALICE Run1 data file

MPD MC rec file

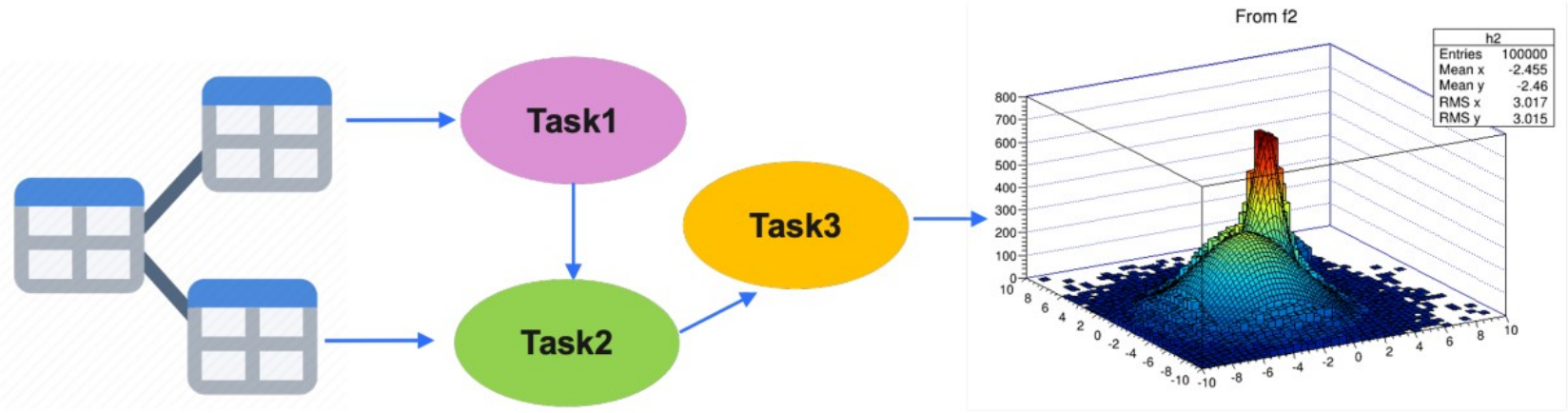


ALICE Run3 data file



O2 Analysis Framework

- General structure of Data processing



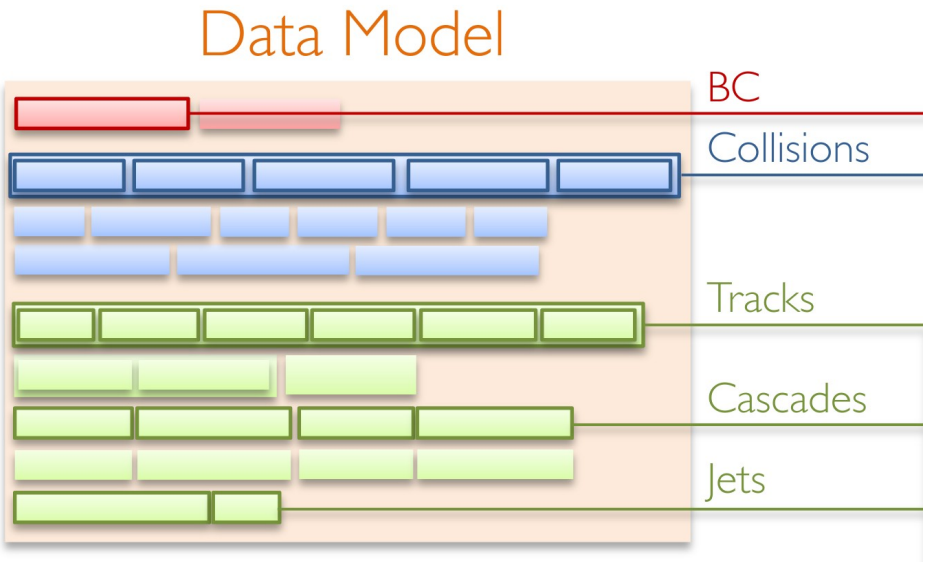
Interconnected tables
Based of Apache Arrows

User Tasks
workflows~wagons

AnalysisResults.root
+AO2Ds (derived data)

O2 Analysis Data Model

Apache Arrow Tables



Each analysis task is an executable → All the required are run in command line with pipe “|”

Plenty of Helper tasks → Produce required data tables on the fly

O2 Analysis Model: Types of Wagons

User wagons:
Spectra
Correlations
etc.

Core Service wagons: helpers, dependencies of user wagons:
Centrality
Event Selection
Multiplicity
Timestamp Creator
Track Propagation
etc.

Wagon – analyzer
Creates and stores user defined analysis histograms

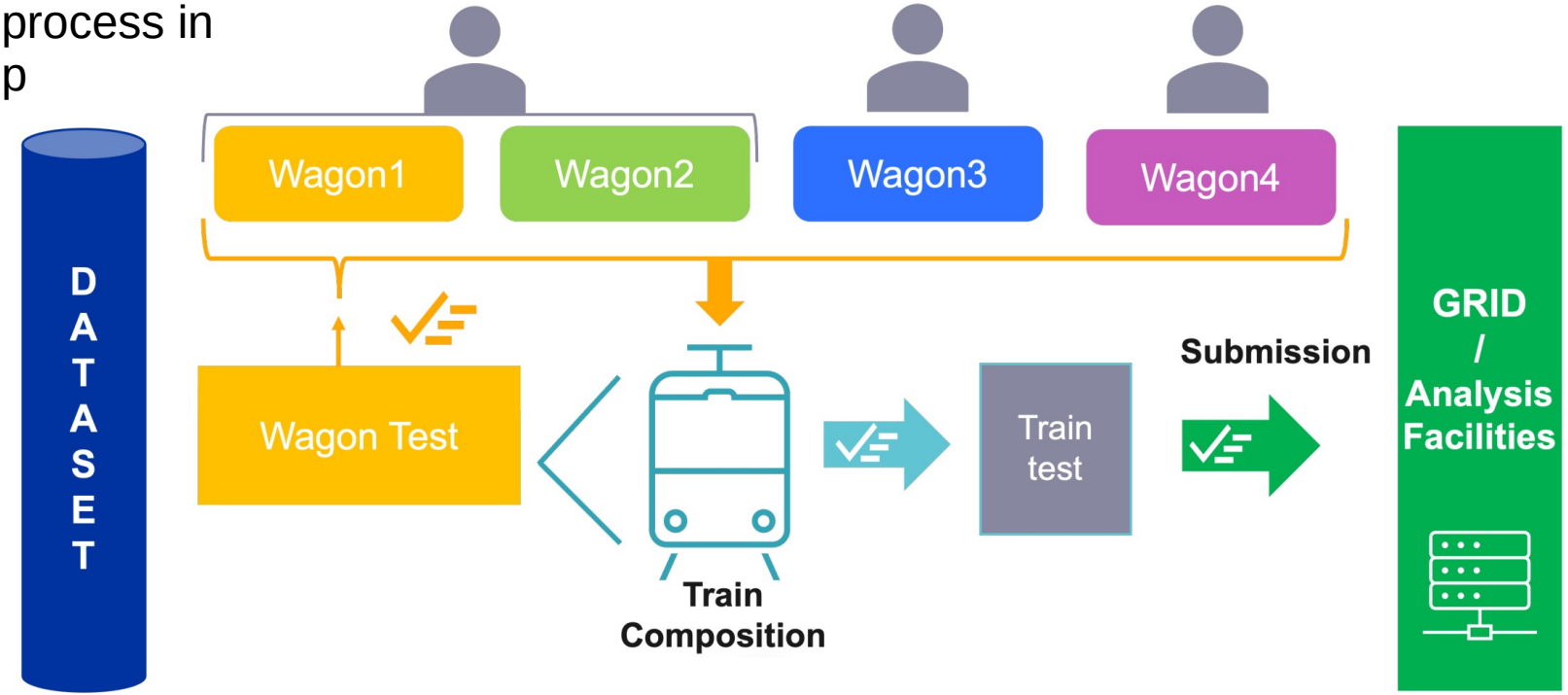
Wagon – producer
Mostly intended for generation of the derived data, which is used by other wagon or saved into AO2D files

Wagon – reader
Analysis of the derived data

Wagon with parent level access
Can have access to the parent data

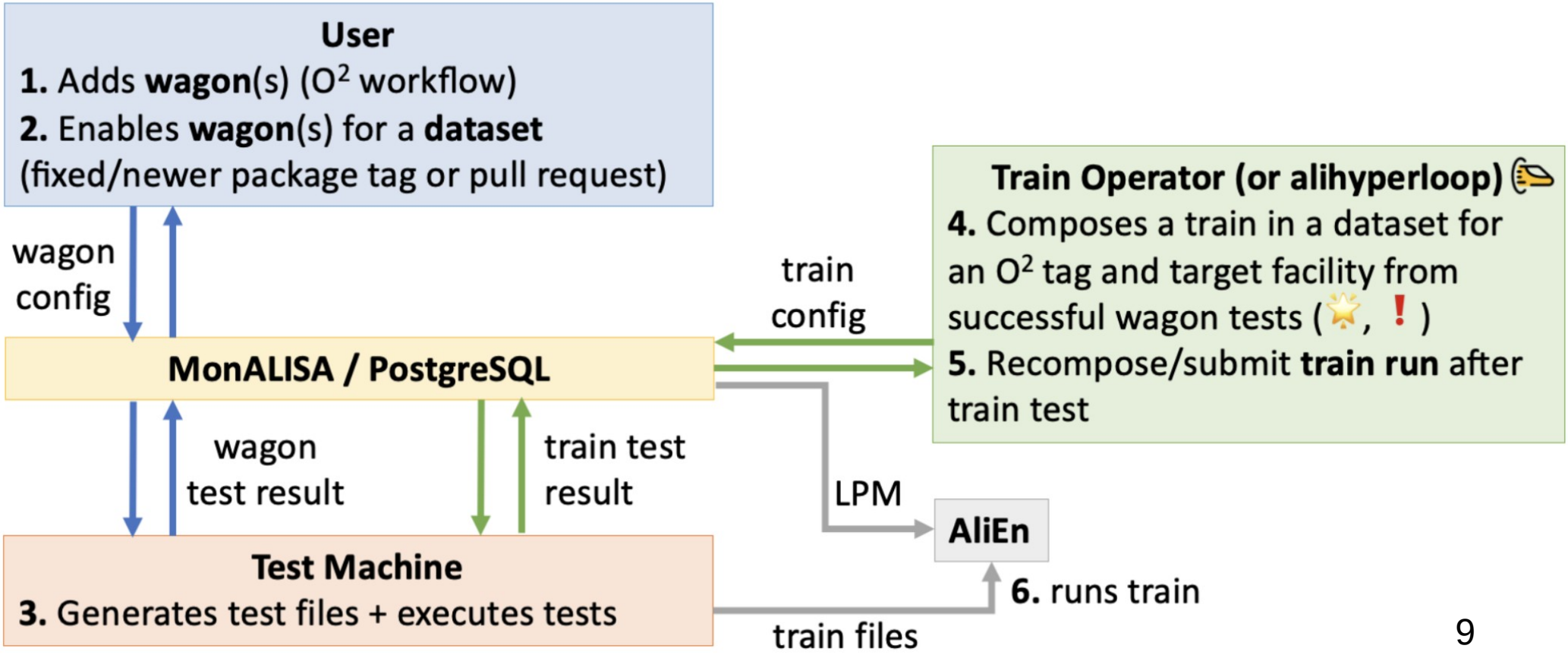
Hyperloop trains

Analysis process in Hyperloop



Config of each wagon is saved and stored in JSON file
Configs of all wagons are merged into general train's config

Process to submit hyperloop train



Comparison of Hyperloop trains and LEGO trains (Run 2)

ALICE LEGO train

- Analysis train framework for Run 2
- Analysis code is contained in an AliEn package, AliPhysics, and delivered via CVMFS
- Trains are defined per Physics Working Group (PWG), data type and collision system (~100)
- Analysis tasks (wagons) using the same dataset are run together
- Requires train operators (per PWG) to test, compose and submit train runs
- Main workhorse for Run 2 analysis:
  - 2020: 16 000 trains, 172 million Grid jobs

ALICE Hyperloop

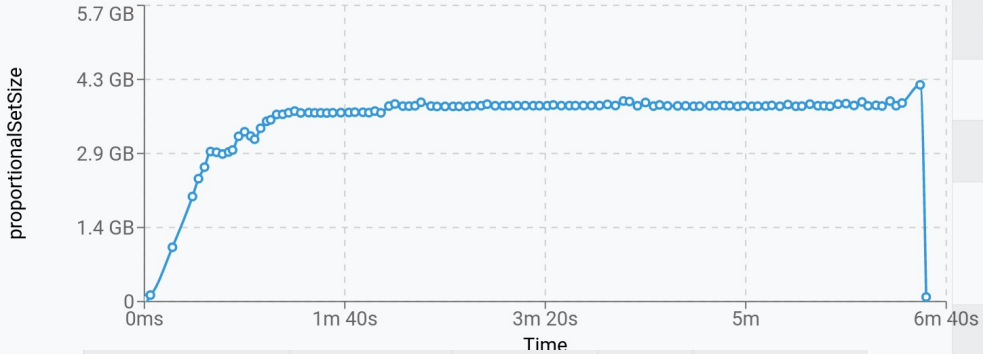
- Analysis train framework for Run 3
- Analysis code is built into O2Physics available on CVMFS
- Advanced web-interface (frontend: React.js)
- Unified trains throughout PWGs
- Personalized user and operator interfaces
- Immediate and automatic wagon test
- Automatic train submission under certain defined conditions
- Wagon and dataset bookkeeping
- Usage in 2023: ~ 9000 trains, 24 mln Grid jobs
  Usage in 2024: ~ 24000 trains, 76 mln Grid jobs

Wagon and train run test performance results

Train run 117133

General | Test | Submitted jobs | Grid Statistics | Wagon resources | Merged Output | Clone

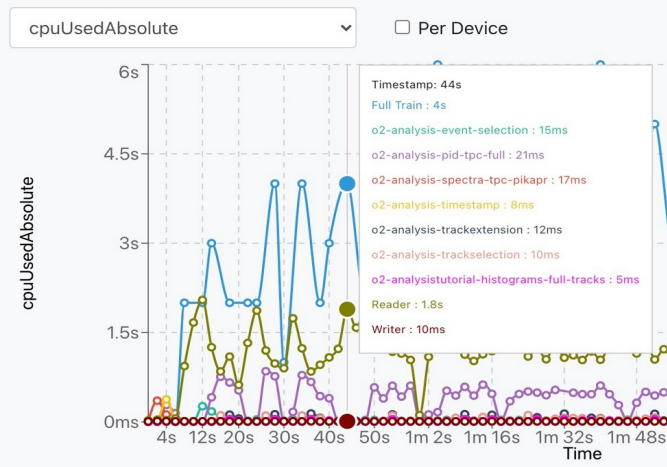
Full Test | Per Wagon | Graphs



Wagon		PSS Memory	Private Memory	CPU Time	CCDB	
					Calls	Transfer size
Reader	Max	1.6 GB	1.1 GB	1m 51s		
	Avg	1.2 GB	794.8 MB			
	Slope	3.9 MB/s	1.8 MB/s			
EventSelection	Max	308.7 MB	120.5 MB	3s		
	Avg	150.4 MB	114.7 MB			
	Slope	-360.1 KB/s	179.5 KB/s			
HistogramsFull	Max	25.7 MB	9.6 MB	2s		
	Avg	19.6 MB	9.5 MB			
	Slope	87.7 KB/s	157.4 B/s			
PIDTPCFull	Max	226.3 MB	158.3 MB	38s		
	Avg	185.5 MB	120.2 MB			
	Slope	368.5 KB/s	11.7 KB/s			

Number of input files	4
Input size	7.0 GB
Output size	4.8 MB
PSS Memory	Max: 3.9 GB Avg: 3.6 GB Slope: 2.4 MB/s
Private Memory	Max: 3.4 GB Avg: 3.2 GB Slope: 1.2 MB/s
Timing	CPU: 9m 57s Wall: 6m 29s
Throughput	9.2 MB/s/core
Expected resources	19h 49m

Per wagon:



Datasets

*Latest change by **rcruceru** at **02 June 2023 at 10:20:45 GMT+3***

Edit dataset

LHC22f_pass4 (DATA)

https://alice.its.cern.ch/jira/browse/O2-3790

History (bookkeeping)

Options Learn more

✅ Activated ✅ Run final merging over all runs in this dataset ❌ Dataset sampling

Linked Datasets:

- Dataset is updated by *rcruceru* — *02 June 2023 at 10:20:45 GMT+3*
- Dataset (production) is created by *rcruceru* — *02 June 2023 at 10:20:09 GMT+3*
  - Dataset production **LHC22f_apass4** is created
    - Mergelist **all** of **LHC22f_apass4** production is created
- Dataset is created by *rcruceru* — *02 June 2023 at 10:19:51 GMT+3*

Analysis Facility Staging Learn more

Dataset size: 661.7 GB **File Pattern:** AO2D.root

Not staged

Dataset size varies from few GB to several PB

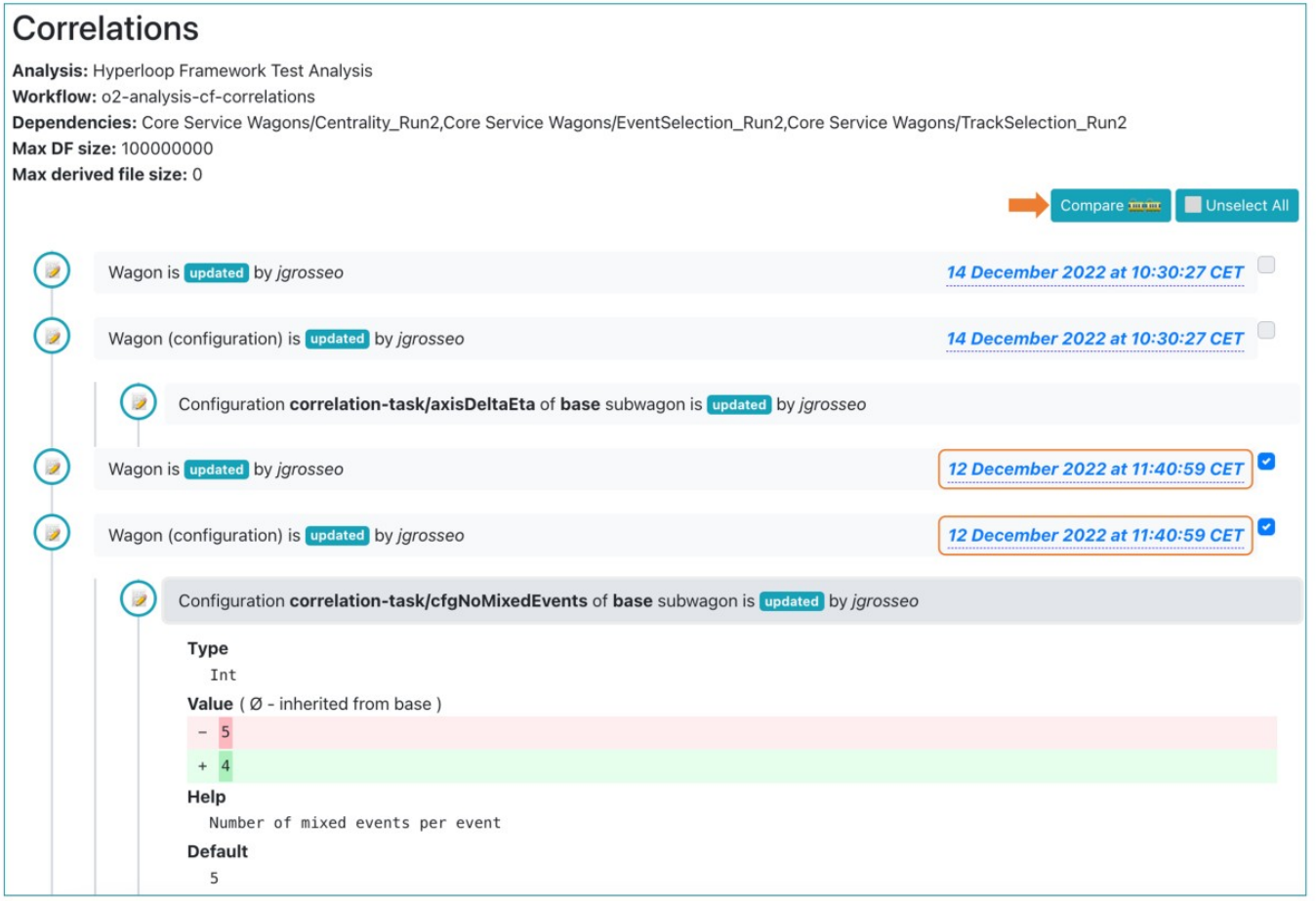
Automatic Train Composition Learn more

Automatic train composition: Scheduled **Maximal CPU time in days:** 550 **Maximal trains per analysis per week:** 14

Composition schedule (CET): Monday - 03:00 | Monday - 15:00 | Tuesday - 03:00 | Tuesday - 15:00 | Wednesday - 03:00 | Wednesday - 15:00 | Thursday - 03:00 | Thursday - 15:00 | Friday - 03:00 | Friday - 15:00 | Saturday - 03:00 | Saturday - 15:00 | Sunday - 03:00 | Sunday - 15:00

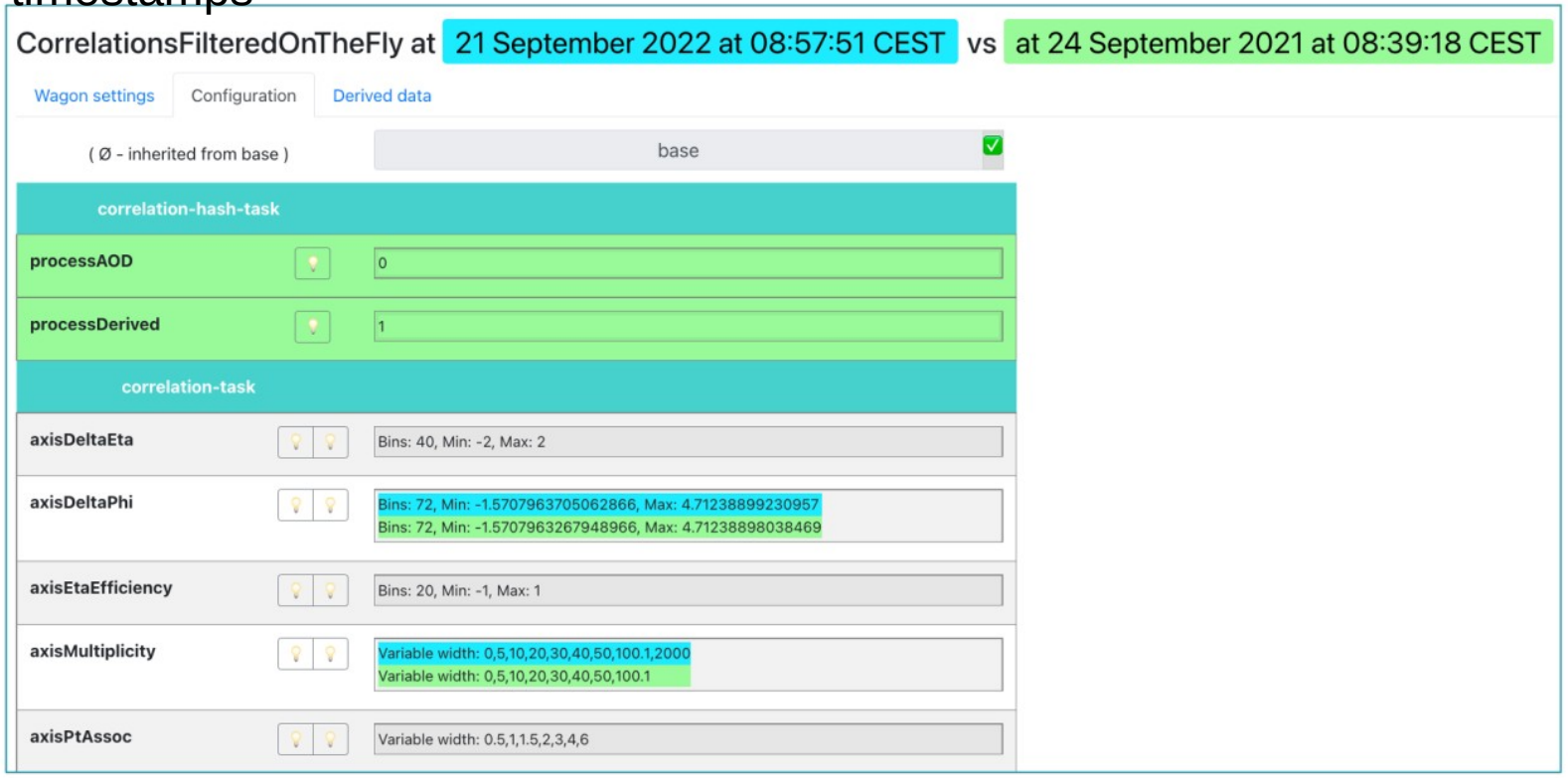
Bookkeeping

Wagon changelog



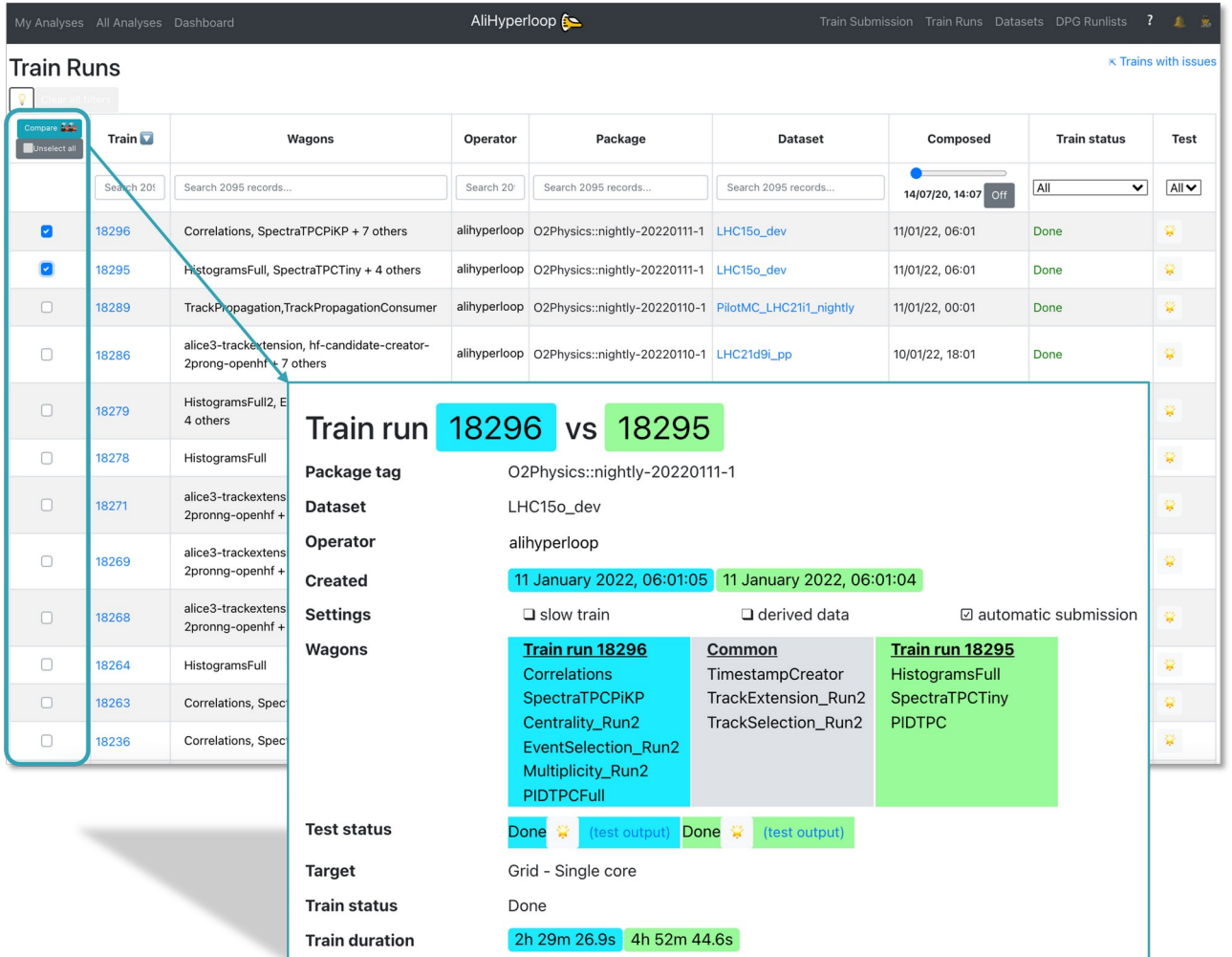
Bookkeeping

Wagon comparison at different timestamps



Bookkeeping

Train comparison



Roles of Analyzer and Operator

ANALYZER

My Analyses

All Analyses

Dashboard

- Creates and configure wagons
- Runs wagon tests
- Studies test results
- Makes use of automatic train composition or ask to compose train
- Studies the resource consumption
- Stores derived data to be used in subsequent trains
- Makes use of history and statistics views

OPERATOR

Train Submission

Train Runs

Datasets

Derived Data

DPG runlists

Trains with issues

- Runs the system on a daily 24/5 basis
- Ensures efficient usage of the resources
- Follows up on overall system status
- Investigates issues and delegate to experts
- Responds to user requests
- Submits trains to the Grid or AFs
- Manages datasets
- Creates datasets of the derived data

- PWG Convener: approves long trains (> 200 Tb)

Types of the derived data

- Slim derived data (<10GB, single usage, no dataset creation)
- Standard derived data (>50 GB, derived datasets created):
  - Femptoscopic correlation datasets
  - reduced CFCollisions and CFTracks for correlation analysis
  - muon datasets
  - electormagnetic probes datasets (e+ e-)
  - multiplicity studies datasets
  - strangeness
  - etc
- Linked derived datasets:
  - Mostly for heavy flavor usage: indices for HF prongs and vertices of cascade decays
  - To be processed together with the parent dataset

Current status

- Hyperloop is in production since early 2022.
- Run 3 data and MC are available for the analysis.
- All Run 2 data and considerable amount of MC data converted to AO2D format and available on Hyperloop
- Operator support on a daily 24/5 basis by four institutional clusters in different timezones

NISER	2:00-9:00 UTC
St. Petersburg / Münster	9:00-13:00 UTC
Brescia/Pavia	13:00-17:00 UTC
US cluster	17:00-1:00 UTC

- ~1300 datasets available (including derived data)
- ~450 users of hyperloop system
- More than 400k wagon tests done, ~52100 trains run on Grid or AF
- The average completion time of the trains is between 8 and 16 hours
- Activity in Hyperloop increased significantly within the last three years

Analysis in MPD NICA at JINR, Dubna

- Code organized in the central repository mpdroot hosted at gitlab
- Compiled code is distributed via CVMFS, for Centos7 and Rocky Linux 9
- Production data available on several facilities:

  -- dedicated NICA Cluster with Slurm on board

  -- HyberLIT computing infrastructure

  -- “Govorun” supercomputer (named after Nikolay Nikolaevich Govorun)

  -- Dirac distributed infrastructure (like WLCG) (similar but extended Dirac middleware is used also by LHCb)

  - X509 certificates, VOMS authentication
  - In MPD not for analysis for now, mostly for Monte Carlo production

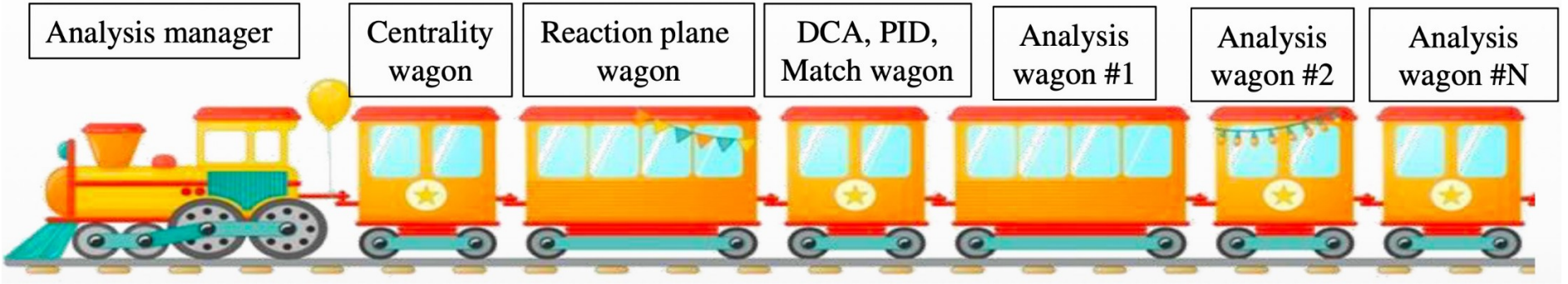
Analysis trains at MPD NICA at JINR, Dubna

- Analysis Train became a new standard for physics (feasibility) studies in MPD

Requirements for the analysis framework:

- Consistency of approaches and results across the collaboration – robust crosscheck of the analysis
- Ability to easily implement analysis in the framework – modular structure of the software, code standardization
- Easy data storage and reduced number of I/O operations – execution of the modules in one sequence

Solution: Analysis Train



First Analysis Train runs started in September 2023 – regular runs on request
Continuous development

Formative function of data analysis trains

• Organized analysis requires to write the code carefully, compatible with standards, efficiently enough and documented (at least minimally)

• Code has to be committed to the repository, pass automatic checks and be approved by corresponding code owner (PWG convener for example)

• Analyzers has to be disciplined to be ready for the train start, to plan work accordingly (in Hyperloop trains are very regular though, with daily software tag)

• Example: it is possible to process the Run3 data in ALICE as single user, all accesses are granted, but just there is no instruction how to do it.

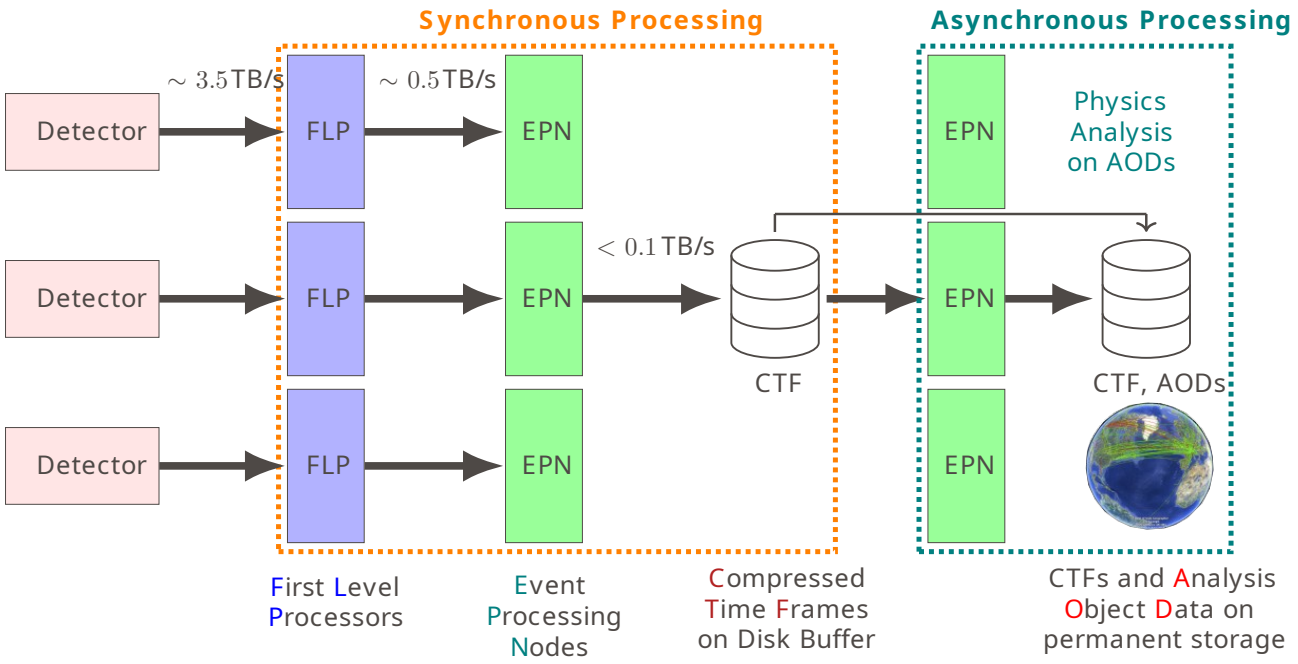
- users are encouraged to commit their code and run everything as a train.

Note more than 50% trains are single user trains in hyperloop currently.

Thank you

Backup

LHC Run 3 challenges of Data Processing



1 month of Pb–Pb data would produce several PB of final AO2Ds

WLCG infrastructure and AliEn framework

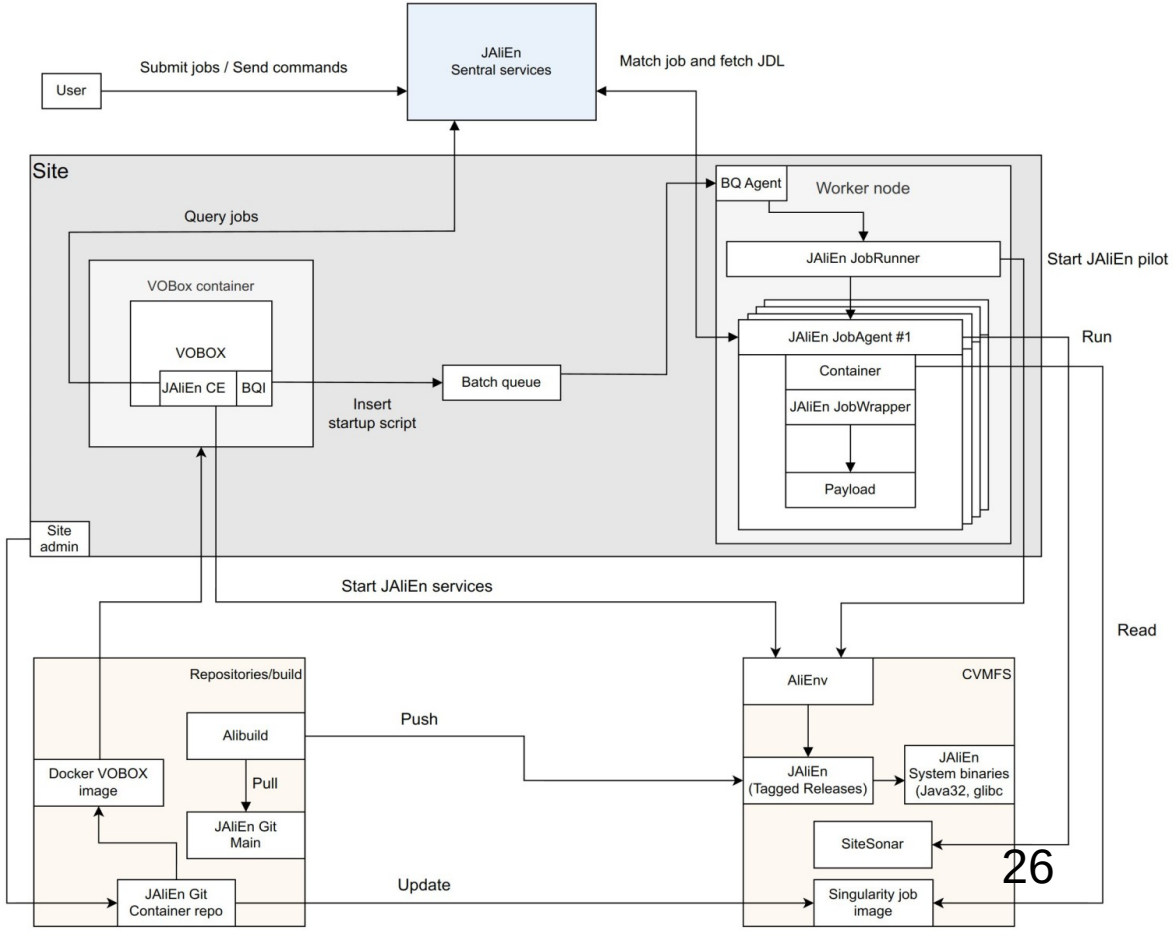
- JAliEn Middleware

  Grid Framework - combination of a Web Service and Distributed Agent Model. Data storage and job management.

- CVMFS central repository

  ALICE analysis and required supplementary software delivery to run in Grid sites in containers. Integrated with build system for continuous deployment.

- Analysis software:
  - User
  - Centralized (Trains)
  - Data and MC production



Analysis in JINR DIRAC

Dirac usage (2023)

Experiment	First usage	Jobs done	Consumed CPU, HS06 days	Consumed Walltime	Data generated, TB
MPD	Aug 2019	1.07 M	5.47 M	840 years	330
Baikal-GVD	Oct 2020	123000	590 k	90 years	40
F@H	May 2020	13000	137 k	23 years	n/a
BM@N	Jul 2021	22000	170 k	30 years	18
SPD	Nov 2021	20000	78 k	18 years	43

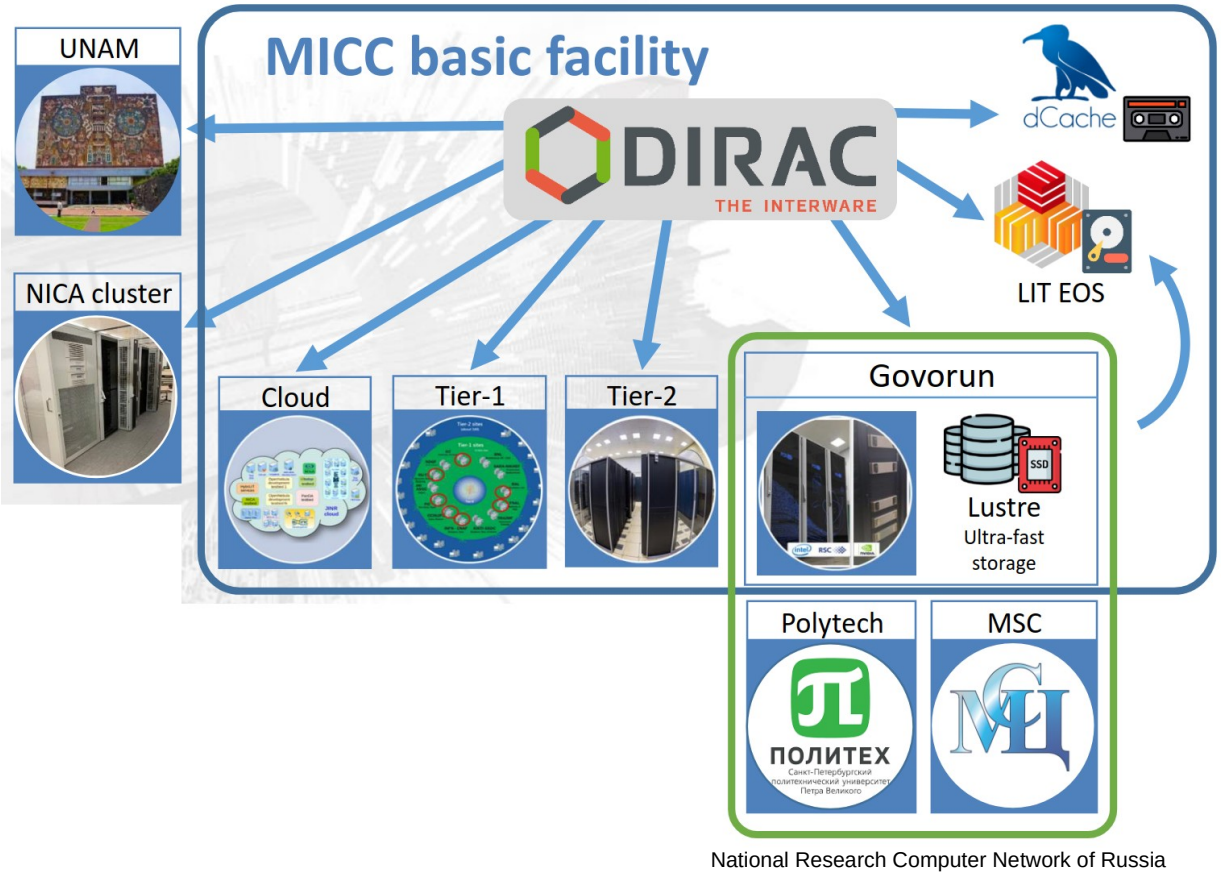
Table 1: Overview of all DIRAC users

Vladimir Korenkov et al 2023
J. Phys.: Conf. Ser. 2438 012029

Analysis in JINR DIRAC

- NICA offline cluster 300 cores (limit for users)
- GOVORUN up to 3260 cores
- Tier1 1400 cores
- Tier2 1000 cores
- Clouds (JINR and Member States) 70 cores+
- UNAM (Mexico University) 100 cores
- National Research Computer Network of Russia (now resources from SPBTU and JSCC) 672 cores – New resource

File Catalog have size 2,3 PB.



max 3500 simultaneous jobs

HNATIC Slavomir
https://indico.jinr.ru/event/3505/contributions/22209

Data types to process

- MC True only data
- MC Reconstructed + MC True data
- Real data RAW
- Real data Reconstructed
- Derived data: MC True+Rec
- Derived data: real data
- Service data – metadata, conditions database, configurations, settings, etc

Data types to process

- MC True only data
- MC Reconstructed + MC True data
- Real data RAW
- Real data Reconstructed
- Derived data: MC True+Rec
- Derived data: real data
- Service data – metadata, conditions database, configurations, settings, etc

Data storage (packing) ↔ Analysis framework

- Old school: Fortran dominates
  - PAW (Physics Analysis Workstation) 1986
- CERN Root 1995 (René Brun, Fons Rademakers)
  - command line C++ interpreter (CINT in version 5, cling in version 6)
  - Object-oriented
    - Encapsulation, inheritance, polymorphism
    - ROOT's C++ object serialization:
      from memory to disk and back

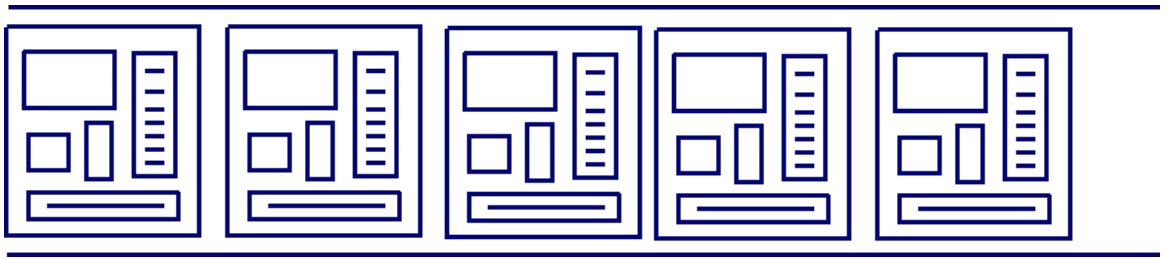
CERN ROOT C++ since 6.0 version

- Switch from Cint to Cling

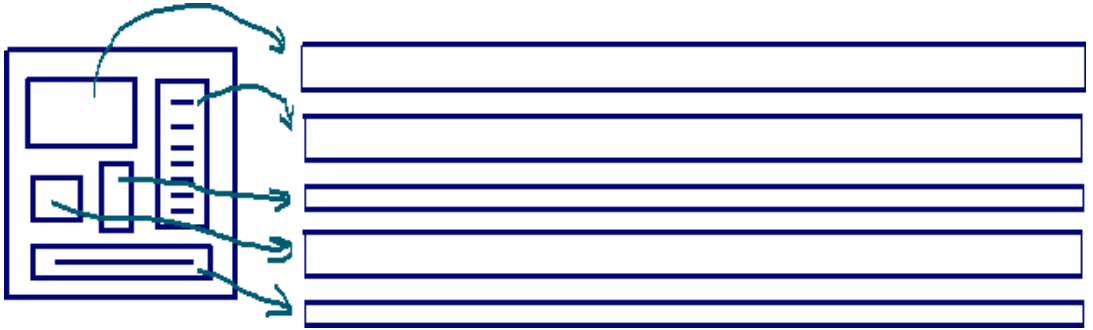
- More stable, more compatible with C++
- Compatibility with modern standards C++11/14
- Since then compatibility with new standards is ensured
- Write code in ROOT using C++ syntax
  vs
  write C++ code using ROOT as a library

Data storage: AoS or SoA

- Array of structures (AoS)

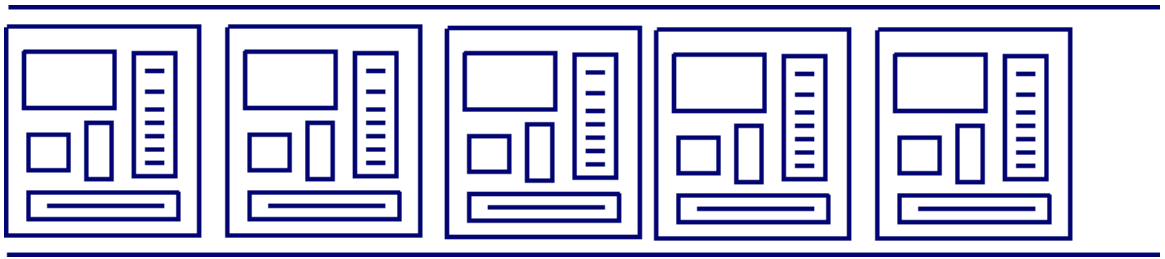


- Structure of arrays (SoA):



Data storage: AoS or SoA

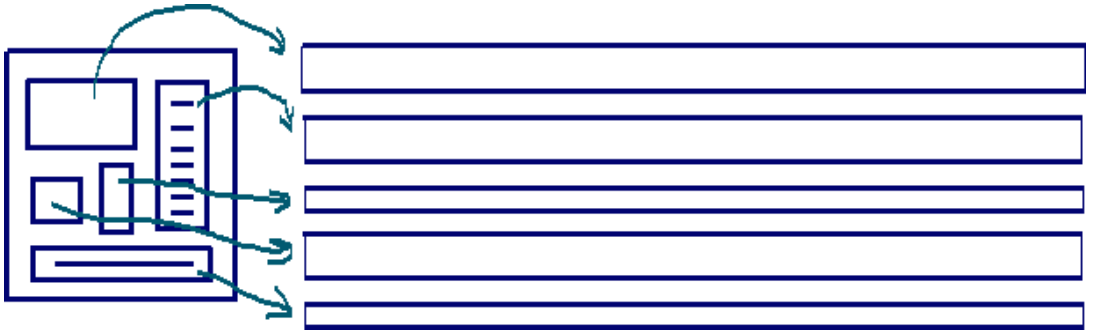
- Array of structures (AoS) – mostly all LHC Run 1+2 data



  - Context switches in processing, memory access not effective
  - Parallelization not effective
  - Long-term storage has problems, compatibility issues

Data storage: AoS or SoA

- Structure of arrays (SoA): - LHC Run 3 data at ALICE (AO2D)



- Uniform objects to read
- Memory access effective
- Parallelization is effective
- Good flexibility in replacement and combining data