

ROOT Q1 2024 Report

Danilo Piparo (CERN, EP-SFT)

3-04-2024



About This Meeting

- ▶ CMS suggested at the [ROOT-Experiments meeting about 2024 plans](#) to report quarterly on the completion status of the Program of Work (PoW)
- ▶ We welcomed the idea, and that's what we'll do today
- ▶ We'll also profit from your presence to present other metrics, broadcast some news and discuss other matters, among which the release planning for 2024
- ▶ Questions, feedback, and comments are welcome during the talk or at its end

We also have questions, e.g.:

1. We think the communication with experiments and users is working well: can it be improved?
2. We think the process by which users and experiments are involved in the ROOT planning and priority settings is well structured and clear: can it be improved?

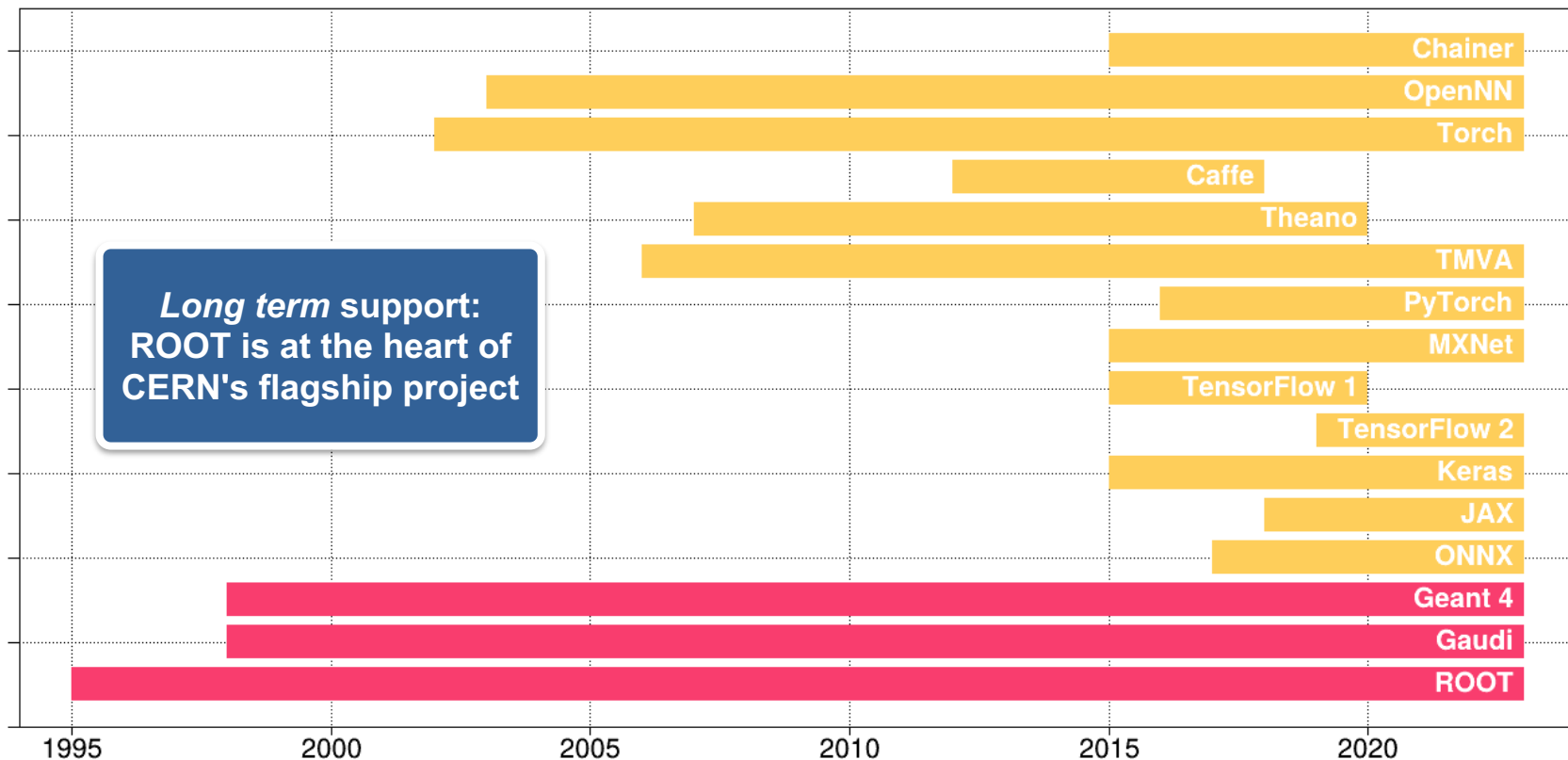


Provide a unified software package for the storage, processing, visualisation and analysis of scientific data that is reliable, performant and supported, that is easy to use and obtain, and that minimises the computing resources needed to achieve scientific results.

The success of experiments and all ROOT users at large is our priority



HEP Software Support Timeline



Plot inspired by [M. Mazurek](#)

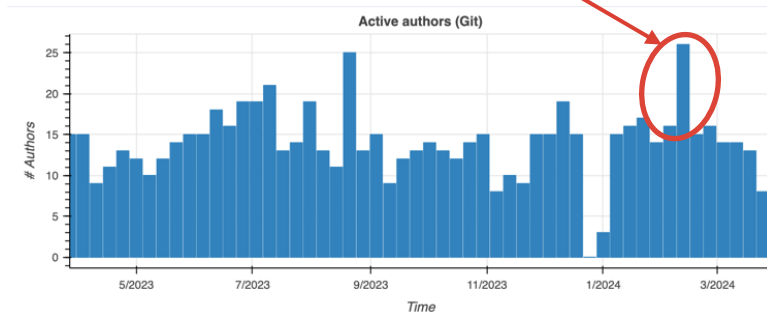


Q1 Metrics



ROOT Community and Development

- ▶ ROOT is an open source project, supported by a lively community
- ▶ ROOT Forum: 9h on average to obtain a first response (20h in 2023) -2.6k posts (11.8k in 2023)
- ▶ **About 15 active developers/contributors at any point in time** (same level of 2023)
- ▶ **695/528 PR opened/closed, 1 day median to close a PR**
 - 1740/1698 in 2023

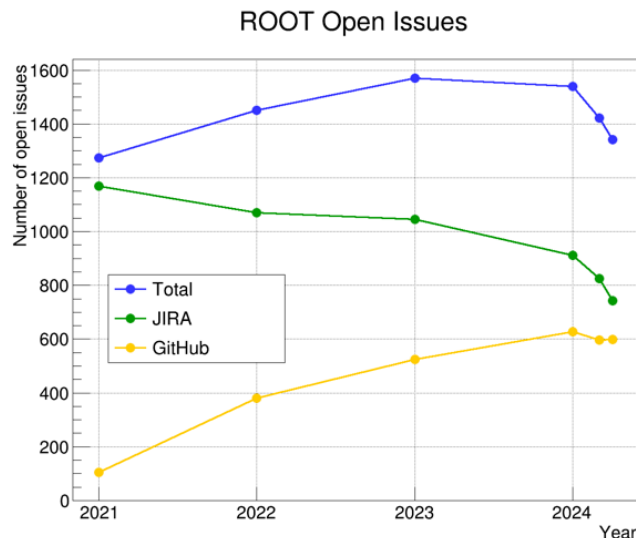




Open Issues

- ▶ Strong focus on reducing number of open issues
- ▶ Backlog reduction is implicitly *part of the PoW*
- ▶ Great boost given by the ROOT community and team at the [1st ROOT Hackathon: the Fixathon](#) (14-15 February)
- ▶ 195 new issues created, of which 98 closed
- ▶ **394 closed in total**
- ▶ **13% reduction of # issues during Q1**

	JIRA	GitHub	Total	Notes
Dec-20	1169	104	1273	
Dec-21	1071	380	1451	
Dec-22	1045	525	1570	
Dec-23	912	627	1539	
Feb-24	826	596	1422	54 JIRA issues migrated to GH
Mar-24	739	601	1340	10 JIRA issues migrated to GH



1st ROOT Hackathon: The Fixathon



Are you interested in Open Source scientific software?

Have you always wanted to contribute, but never knew where to start...?

...Or do you already have some (or many) commits under your belt and would like to add more?

Join us for the 1st **ROOT FIXATHON**

February 13th & 14th 2024 • CERN IdeaSquare
Homecooked lunches included!



<https://cern.ch/root-fixathon>



ROOT
Data Analysis Framework IdeaSquare

- ▶ Took place at CERN, Idea2 on February 13-14 ([link](#))
 - ▶ ~30 people, from early career physicists to senior scientists,
 - ▶ Onboard early career colleagues teaching how to contribute to a large open-source scientific software
 - ▶ The main objectives were all achieved:
 - Close as many ROOT issues as possible
 - >70 issues closed (for comparison, 441 were closed in total during 2023), 17 solutions proposed, work started on another 23
 - Connect further with ROOT community, inclusively
- 2nd ROOT Hackathon 25-26 November 2024 at Idea2: save the date!** More details will be circulated in the next few months.





Spotlight on the PoW



ROOT's 2024 PoW: Timeline

- ▶ November 2023: [PoW Jamboree](#)
 - New (Axel's idea): presentations about plans of different ROOT focus areas, prepared and delivered by team members *not* directly responsible for them: incentive to motivate and pre-discuss future directions.
- ▶ 15th January: [SFT Special Group Meeting – PoWs 2024](#)
- ▶ 23rd January: [ROOT Funders Meeting – PoW 2024](#)
- ▶ 24th January: [ROOT-Experiments Meeting – PoW 2024](#)
- ▶ 30th January: [Meeting with CMS Common Analysis Tools group \(CAT\)](#)
 - New: get feedback from teams outside of the traditional area of central Offline and Computing coordination areas. Repeated the success with the Trigger Study Group area a few days later
- ▶ 1st Feb: [AF Feedback on SFT PoW](#) (thanks to Giulio Eulisse for consolidating all the feedback!)
 - Will focus on this feedback in the next slide

Very useful exercise: at each of these steps, the trajectory was refined and the PoW improved

- ▶ The items in the PoW are now 69 (they were 61 at the beginning)



Feedback from the AF Talk

Excellent presentation, useful to obtain consolidated feedback from all experiments for all SFT projects at once

- ▶ Example of good feedback for all projects: "good to have an optimistic spin on future plans, it would be good to have some critical look at the past"
→ **more about this in the next slides.**

Executive summary for ROOT and initial response:

- ▶ *Copy-less bulk reading (ALICE). Performance is the feature sometimes.* → **Added to the PoW and [PR already merged](#)**
- ▶ *Evolution path for polymorphic types (CMS, see also November plan at ROOT workshop)* → **Added to the PoW, work started**
- ▶ *Unaligned friends (LHCb)* → **Added to the PoW**
- ▶ *ROOT 6.32.00 release should be early (i.e. CMS: August, LHCb: September, ALICE: November) to be adopted for 2025 datataking.* → **A proposal to address this request will be made today at the ROOT meeting**
- ▶ *Everyone welcomes the rebasing on clang-repl. In general, reducing customisation on top of CLANG / LLVM is very welcome.* → **Ongoing work to rebase to llvm 18 and to standardise llvm update procedure.**
- ▶ *Security of the Web GUI: it would be good to see explicitly mentioned a path to a more secure solution (e.g. the one Jupyter uses).* → **Changes already merged to master, targeting 6.32**
- ▶ *ALICE: memory overhead of cling should still be on the radar and it would appreciate progress on the TBufferFile 1GB limit.* → **Already in the PoW, strategy forming, discussed with [priority at I/O meetings](#)**
- ▶ *Everyone feels that Windows support should be descoped to "Community Effort".* → **Took good note of this item, still unclear how to find actor(s) in "the Community" that can sustain the effort**



Measuring Progress: PoW Completion

- ▶ Feedback from ROOT-Experiments meeting: it would be useful to provide feedback about the PoW completion every quarter.
- ▶ **Needed to find a good way to address that keeping Project Management formalism as lightweight as possible. Our solution:**
 - **Keep 2 priority levels: 1 and 2** (+ "stretch goals": the items we will address if an injection of effort, e.g. short term students, takes place)
 - **Assign to every item in the PoW a score: 0 not done, 0.5 partially done, 1 done.** The amount of work needed to complete the items in the PoW varies. We assume, for simplicity, that, overall, it averages out.
 - **Introduce the concept of Extra Items**, i.e. the items not foreseen in the first version of the PoW
- ▶ Remain open to changes in the PoW and prioritise every new item related to 2024 LHC data taking
- ▶ Remain focussed on the ROOT strategic goals

2023 PoW Completion

- ▶ Exercise the framework on the 2023 PoW
- ▶ Completion status:
 - With extra items: 56.4 %
 - W/o extra items: 49 %
- ▶ **CAVEAT:** this is just a number, which does not represent the performance of individuals nor the enormous work done by the ROOT team during 2023
- E.g. see the achievements slides of the [PoW talk](#)

		Priority (1=highest)	Completion Status: 0, 0.5 or 1
	DONE		
	DOING		
	NOT DONE		
Buils and Binaries	CI rewrite including PRs to use GH actions	1	0.5
	Add .deb package generation with CPack	1	0
	Prototype CMake superbuilds	2	0
	Pin install ROOT	2	0
IO and TTree	Address scaling issues with MT-writing to TBufferFile	1	0.5
	TBufferFile = 1 GB	1	0
	Schema evo improvements	2	0.5
	Beta release of lossy compression + incorporation in RNTuple	2	0.5
	Support for STL collection of std::array	2	0
RNTuple	Bulk I/O API and RDataFrame connection	1	1
	Late schema extension	1	1
	(Prototype merged, missing support for edge cases) Support for merging and chains	2	0.5
	Unaligned friends	2	0
	(Double32, 1, f16 support available, Float16 1 & custom precision missing) Support of lossy compression (low-precision floats)	2	0.5
	(merged, prototype used for CHEP benchmarks) S3 Backend	3	0.5
	Implementation of meta-data API	3	0
	Prototype schema evo based on existing I/O customisation support	3	0
RooFit	Execute LHCb benchmark fits fully on the GPU (result for CHEP2023)	1	1
	Engine for C++ code generation from RooFit model (as in input for AD)	1	1
	Finalise redesigned minimiser interfaces for better support of automatic differentiation (Clad)	1	0.5
	Further consolidate JSON standard, joint target with ATLAS: publish joint tW and tH multilepton full Run-2 analysis	1	1
	Support for likelihood parallelisation with new test statistics and improve scheduling of gradient parallelisation	2	1
	Consolidate new test statistics classes: de-duplicate common code, unify interfaces, enable vectorised/gpu + MP fits, etc.	2	0.5
	Stabilise and improve the code, and speed up the HistFactory	2	1
	Pythonise RooWorkspace factory language	3	0
	Creates more ROOT benchmarks that compare RooFit also with other fitting tools to get a better overview on fitting tools in HEP	3	1
RDataFrame	Bulk processing, also with RNTuple readers	1	0.5
	Default values for missing columns	1	0
	Distributed support for RDatasetSpec	1	0
	Prototype CUDA kernels in RDF	2	1
	Varied Snapshot	2	0
	Live histograms (streaming results as they come from the mappers)	2	1
Math	Apply several improvements in Minuit2 (e.g. Fumli algorithms)	1	1
	PARTIALLY/Complete Pythonizations of Histograms and Graph classes	1	0.5
	Benchmark Minuit2 against optimisers of scipy and eventually integrate some of those algorithms in ROOT	2	0.5
TMVA	Batch generator integrated with RDataFrame to train ML models	1	1
	Add support for SOFIE for inference of GNN	1	1
	Consolidate SOFIE adding support for missing ONNX operators according to user requests	1	1
	Make SOFIE inter-operable with HLSAMEL	2	0
	Consolidate RBOT (fast BOT inference)	2	0
Visualisation and UI	REve - RenderCore using GPU, window manager, drop ROOT: Experimental namespace	1	0.5
	TWebCanvas - use by default as with TBrowser	1	1
	Optimise object Paint methods - avoid gPad as much as possible	2	0.5
	Support RWebWindow in JupyterLab - make it fully interactive	2	0
Interpreters	Seamlessly translate PyROOT-bound C++ code via Numba	2	0.5
	Risc-V support for Cling	1	1
	Cling: Q2 for non-interactive ROOT on Linux and Mac	1	0
	Reduce dependency on precise version of SDK headers, esp. For macOS	2	0
Extra Items	5x speedup of Import ROOT	1	1
	RNTuple support of std::unordered_set/map	1	1
	Comprehensive RDF scaling tests on a single multicore node and in distributed mode on many multicore nodes	1	1
	RNTuple Inspector	2	1
	RNTuple support for all ATLAS data tiers persisted so far with TTree	1	1
	RDataFrame analysis chains of RNTuple datasets	1	1
	Migrate ROOT's LLVM to LLVM 16 and reduce the number of custom patches from 86 to 56	1	1
	Deprecation of Python 2	2	1
			overall: 49.0
			overall Extra Items: 56.4



2024 PoW Completion

Category	Description	Priority	Completion Status:	0, 5 or 1	%
DONE					
PARTIALLY DONE					
NOT DONE					
Builds	pip install ROOT for some selected platforms	1	0.5		
and	Complete transition to GH Actions, adding GPU runners	1	0.5		
Binaries	Reduce number of services hosted by root.cern with a combination of CERN IT central services	1	0		
	Win: Replace Debug builds with ReleaseWithDebInfo in the CI	1	0		
	Optimise dictionary dependencies to minimise build real time	2	0		
	Win: Add support for Ninja	2	0	16.7	%
I/O	Support std::variant, both in TTree and RNTuple (CMS)	2	0		
and	Support writing objects larger than 1GB (TBufferFile > 1 GB, ALICE)	2	0		
TTree	Complete schema evolution improvements	2	0		
	Ensure consistency of std::int types across ROOT I/O	2	0		
	Address residual scaling issues with MT writing	2	0	0.0	%
RNTuple	Support for chains of datasets and merging	1	0		
	Complete implementation of datasets chains and merging	1	0.5		
	Limit testing in collaboration with CERN IT	1	0		
	Follow-up on API review by HEP-CCE	1	0.5		
	Implement unsplit ("blobified") encoding	1	0.5		
	Support for unaligned friends	1	0		
	Further develop support for lossy compression with low-precision floats	2	0		
	Design compression dictionaries and understand implications for the specification	2	0		
	First implementation of highly-scalable parallel writing	2	0.5		
	Organise a Design Workshop to discuss intra-link events, metadata, native SoA layout for events	2	0	20.0	%
RootFit	Workshop with Experiments: promote features, gather input, speedup integration of RootFit in the existing sw setups	1	0		
	Numeric integrals in n-dim with CUDA	1	0		
	Evaluation of custom user functions in CUDA	1	0		
	Group similar PDFs to speed up evaluation	1	0		
	Make the new vectorized CPU likelihood evaluation interface the default	1	0		
	Reduce JITting time for AD in RootFit	1	0		
	PyROOT: express RooStats configuration with C++-oriented Set* as kwargs	2	0		
	Integration of Fumili in RootFit	2	0	12.5	%
RDataFrame	Put existing bulk processing in prod	1	0		
	DistRDF: reduce memory usage on HTCondor Workers	1	0		
	DistRDF: improve user experience when integrated with notebooks and nb services like SWAN	1	0.25		
	Make the TTree → RNTuple transition transparent for analysers	1	1		
	Further Pythonise the interface	2	0		
	Deliver varied snapshots	2	0.25	25.0	%
Math	PyROOT: better histos and graph interoperability with NumPy and UHI protocol	1	0		
	Histos: advance current RHist implementation to one testable by experiments	1	0		
	Add interface to pass initial error values/cov matrix to Minuit2	1	0		
	Release a library for Lorentz vector computations on accelerators in SYCL	1	0.5		
	Deliver plan and prototype of algorithmic improvements when dealing with param constraints in ROOT's minimisers	2	0		
	PyROOT: Pythonise TF(1,2,3) and numerical algorithms interfaces (e.g. minimisers)	2	0		
	Prototype SYCL kernels to be JITted (see Interpreters objectives)	2	0		
	Histograms: Model and prototype of pipelining GPU histogram filling	2	0.5	12.5	%
ML/AI	Put RBatchGenerator in production	1	0		
	Consolidate RBDT	1	0		
	Support of integration of SOFIE in experiments Fast Simulation pipelines	1	0		
	Add support in SOFIE for NVidia GPUs in CUDA	1	0		
	Continue to add support for the ONNX operators requested by experiments	1	0.25		
	Make HLS4ML interoperable with SOFIE	2	0		
	Streamline ROOT's inference interface, making it able to use models for Python ML frameworks (e.g. Keras/TF) directly	2	0	0.04	%
Visualisation	Automated placement/tune of plot elements, "Auto Style"	1	0		
and	Add missing features of classic graphics to the web-based one	1	0		
UI	Automate web-based graphics test suite	1	0		
	Add residual missing TEvent features to REVe, e.g. digit visualisation and text elements overlay	1	0.5		
	Visualization of flat ntuples using predefined visual summary data structures	1	0.5		
	Improve REVe window manager and browser, polish render engine	2	0	16.7	%

Interpreters	Description	Priority	Completion Status:	0, 5 or 1	%
	Cling: identify potential Cling codebase reductions through the reuse of parts of clang-repl	1	0		
	Cling: copy rebase on top of clang/clang-repl	1	0		
	Migrate PyROOT to the latest Cppyy	1	1		
	Cling: Prototype SYCL support	2	0	25.0	%
Doc and education	Re-evaluate, update, and improve course material, making it more visible and better organised on the website	1	0		
	(Re-)evaluate tuts, eliminating what's outdated, what newer features would benefit from a (better) tutorial, improve visibility	1	0.5	25.0	%
Extra Items	Copyless reading in RNTuple - ALICE	1	1		
	Physics objects representations out of NanoAOD in RDataFrame - CMS CAT	1	0		
	Bulk Processing + GPU offloading for distRDF - CMS CAT	1	0		
	Include the open source Tex Gyre Heros clone of Helvetica in root fonts - CMS CAT	2	1		
	Friend trees and RNtuples - CMS CAT	2	0		
	Support for joins in RNtuples - CMS CAT	2	0		
	Multithreading-friendly interfaces to the histogram types - CMS CAT/TSG	1	0		
	A library of matrix operations that can run on GPUs - CMS TSG	1	0	25.0	%
	overall:			14.9	%
	overall + Extra Items:			16.1	%

- ▶ A large PoW: currently 69 items
- ▶ **Current person power in the ROOT team insufficient to deliver all items**
- ▶ **5 new arrivals foreseen during the year; final completion percentage will depend a lot on these new colleagues**
- ▶ External help, i.e. ROOT community (e.g. experiments, but not only) can make the difference, too

PoW is 16.1 % complete if accounting for extra items



Highlights



ROOT is part of the "Python Analysis Ecosystem". Developments took place with high priority to reinforce this position, for example:

- ▶ **The cppy on which PyROOT relies was updated to its latest version, removing numerous patches** (see [this talk](#))
 - Increased interoperability between C++ and Python
 - About 20 bugs fixed by this upgrade
 - Already tested by CMS (thanks to [S. Muzaffar](#)), changes in *dev3 LCG stack* to be tested by interested parties
- ▶ **A first demonstrator infrastructure to enable "pip install root" was set up**
 - Not ready for users yet, alpha version available during the next quarter

1 FTE to join the ROOT team in Summer to work mainly on PyROOT



A New Approach to CI

- ▶ **ROOT is transitioning its Jenkins-based CI to GitHub**
- ▶ **Fits well Open Source philosophy of ROOT**
 - E.g. test results visible to anybody, further lowers the barrier for external contributors
- ▶ **Integration and testing of the 4 active branches (6.26, 6.28, 6.30 and Main) is now steered with GitHub actions**
 - Jenkins is still running in the background
- ▶ **Version 6.32 will be managed through GH procedures**
- ▶ Expanded the pool of builder nodes. Including 2 mac nodes that are part of the Apple Beta program: catch early problems that could be triggered by the future macos updates.

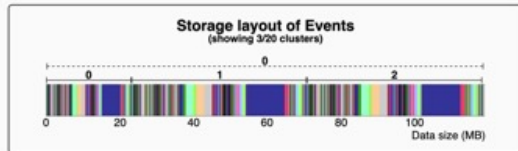
Excellent contributions by ROOT Team members and several ROOT related talks. An (incomplete) selection:

Combined storage information: on-disk layout visualization

We can use the RNTupleInspector to visualize the on-disk layout of an RNTuple

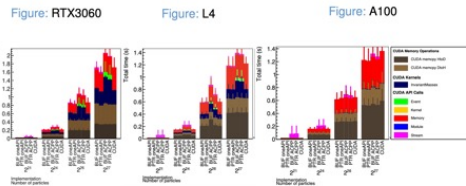
```
lsInspector -DreadStorageLayout("xjetsa_gdf" /* outputPath */, 3 /* nClusters */);
```

Let's first consider the dataset created with the default RNTuple write options:



RNTupleInspector: A storage information utility for RNTuple ACAT 2024, March 11, 2024

NVIDIA GPUs: Total Execution Time Breakdown



Monica Desai (monica.desai@cern.ch) ACAT24 13th March 2024 12/18

Conclusions and Future Works

- Demonstrated Proto-DUNE raw data can be written in GPU friendly format
 - Applied lessons learnt in CCE first iteration to adopt SoA like design to make data GPU friendly
 - Showed the persistence of raw data as SoA in RNTuple
 - Future works
 - Look at further optimization of data models for offloading into the GPUs
- Demonstrated the persistence of CAF data model in RNTuple
 - Future works
 - Investigate I/O support in RNTuple
 - Investigate CAF objects ownership in RNTuple
 - Develop selective reading of CAF objects using RNTuple
 - Write CAF data as SoA
- Examples and test frameworks as deliverables for HEP experiments
 - Simple and standalone examples and frameworks to demonstrate
 - Persistence of HEP data model in RNTuple
 - HPC friendly design of HEP data model and persistence in RNTuple
 - Framework designed for heterogeneous computing architectures

HEP analysis performance evaluation

Evaluating the performance of several High Energy Physics analyses

HNL Run2 search

with distributed features for DAQ compatibility

Analysis made on ROOT RDataFrame [3] v6.27

Preliminary results

The same analysis workflow, running on an increasing number of workers shows a decrease in execution time.

- As expected, low number of workers show a CPU usage saturation;
- For a high number of workers, network access becomes the bottleneck (due to IO access, via protocols like xRootD/WebDAV).

SUMMARY: PROPOSED GW WORKFLOW

- ROOT is very much compatible with GW analysis:
 - ✓ KFX Library, TTimeSeries/TFrequencySeries prototype
 - ✓ TKafoa/oauser/bankier client prototype
 - ✓ TConfigParser/oauser/oauser_tool, oauser_tool prototype
 - ✓ TCoFile / TArchive prototype (understandable file and archive handling)
 - ✓ THDFS parser prototype
- LALSuite Package linkable for direct use.
 - ✓ TStream, TStreamNow, TStreamTruncate (no default static, color and template generation)
 - ✓ TLogsFormat for configuration file parsing
 - ✓ TPhaseChain GWF proprietary format. GMT

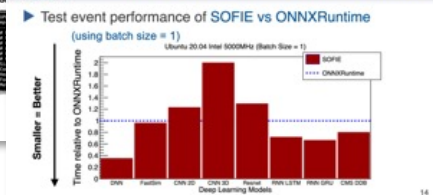
2024 Computing the Wave - Marco Meyer-Conde (ACAT2024) 19

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



CPU Benchmark for Different Models



Distributed AGC with TTree and RNTuple - user side

The only change for the user - the ROOT input file!

```
REMOTE_DATA_PREFIX: str = "root://eospub.lhc.cern.ch/eos/root-eos/AGC/"
REMOTE_DATA_PREFIX: str = "root://eospub.lhc.cern.ch/eos/root-eos/AGC/rtuple/"
```



Releases in 2024



Proposed Release Schedule for 2024

Two releases: one in May (long term support) and one in November (short term support)

May W4 (4th week of May), long term support:

- ▶ Target integration in experiments' stacks for 2025 data taking
- ▶ LLVM16, secure web graphics + more to be decided
- ▶ Branch v6-32-00-patches created this week
- ▶ Release Candidate 1 month before the release

November W1, development release, short term support:

- ▶ Fit all the features we have at that time
- ▶ Branch v6-33-00 1.5 months before: September W2
- ▶ Release Candidate 2 weeks before: October W3
- ▶ The LLVM version is not yet decided, but the plan and decision points ahead of us are clear:
 - We will continue with the endeavour of upgrading ROOT to LLVM 18. Two decision points before October: May and late August. We'll decide if it's worth continuing the effort or if we have hard bugs that prevent us from adopting the version. Three possible outcomes: we stay with llvm16, we move to llvm17, we adopt llvm18.

This schedule is proposed for 2024 to prepare for 2025 incorporating all requests by experiments. The release schedule for LS3 will have to be decided with experiments in the next months.



Conclusions



- ▶ **The PoW is a clear plan ahead of us during the year.** Its conceptualisation started in Nov. `23, refined in multiple steps, involving experiments, funders and users. Issues backlog reduction is considered part of it
- ▶ **We are committed to deliver the items in the PoW, confident in the abilities of the new team members that will join us during 2024 as well as in the help of our community, in line with our Open Source philosophy**

Q1 Stats, in a nutshell:

- ▶ **The PoW is 16.1% complete**, if *Extra Items* are taken into account
- ▶ The number of **open issues reduced by 13% wrt Jan 1st** (1539→1340)
- ▶ The time to obtain a **1st response of the forum was 9h**
- ▶ **695/528 pull requests were opened/closed** (median time to close 1 day)