ROOT Q2 2024 Report

Danilo Piparo for the ROOT team (CERN, EP-SFT)

26-06-2024

This Meeting

- Second of a series: 1st 2024 Quarter report took place on 3-4-2024
- **ROOT reporting every quarter: an idea of CMS**, that we welcomed
- Questions, comments, feedback are welcome at the end or during the talk

As usual, we would have two questions:

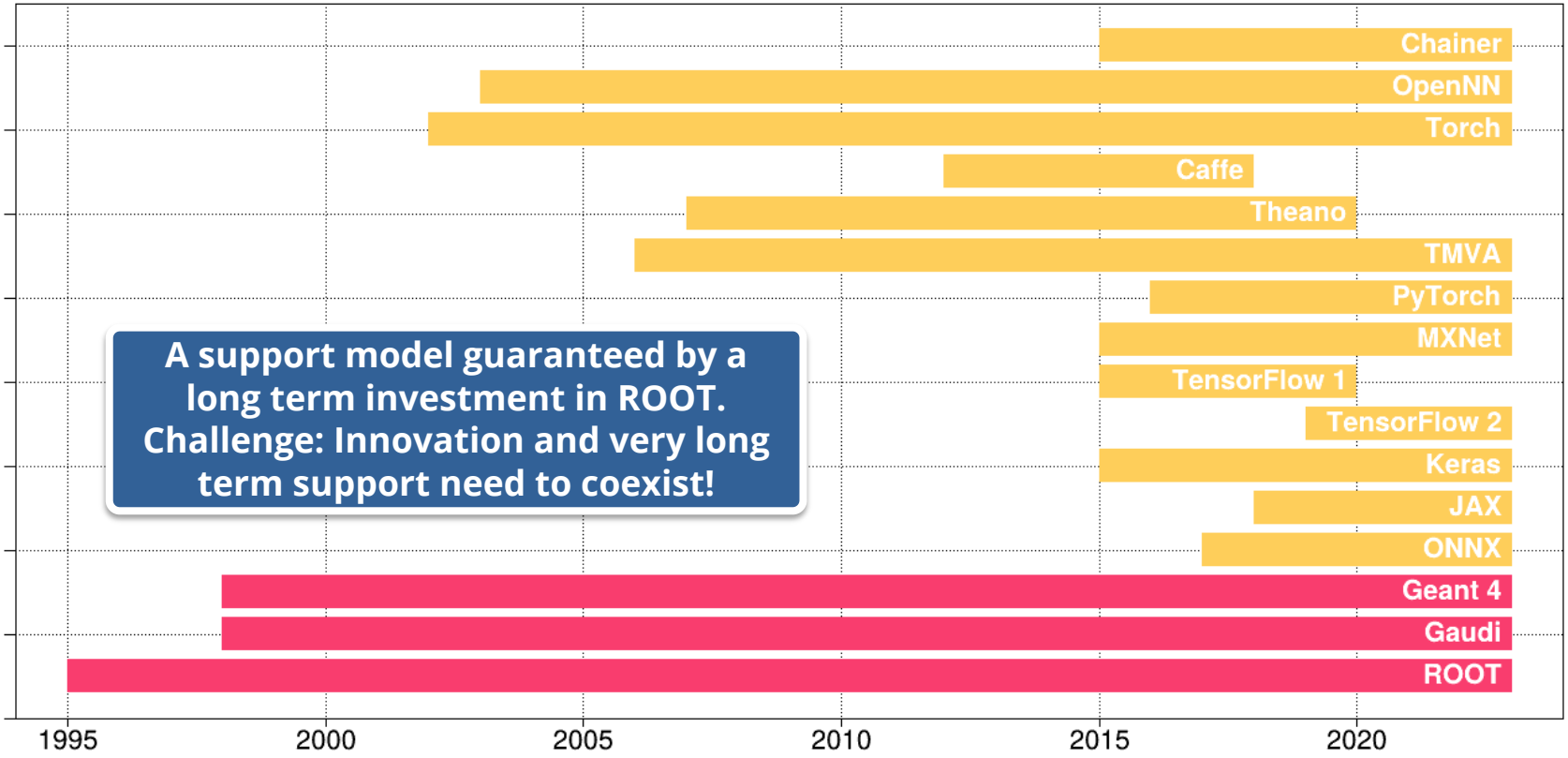
1. We think the communication with experiments and users is working well: can it be improved?
2. We think the process by which users and experiments are involved in the ROOT planning and priority settings is well structured and clear: can it be improved?

ROOT's Strategic Goals

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



A support model guaranteed by a long term investment in ROOT. Challenge: Innovation and very long term support need to coexist!

Plot inspired by M. Mazurek

Who is ROOT?

- **ROOT is its user community, contributors, and developers**
- **ROOT is an international collaboration**, where a sizeable effort is provided by CERN
  - Institutional responsibilities, but also precious contributions coming from the user community!

Team from https://root.cern , PhD students onwards

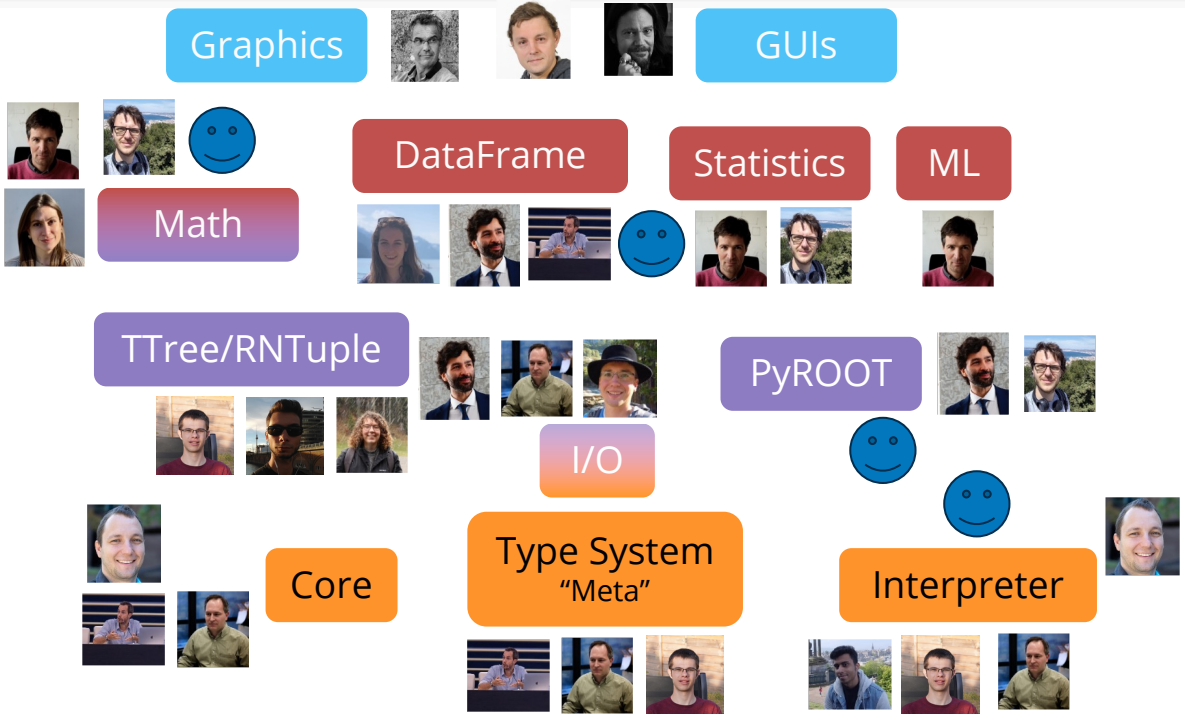
From left to right, starting from the first row (the affiliation is CERN if not specified):

1. Devajit Valaparambil, Bertrand Bellenot, Danilo Piparo, Florine de Geus, Jakob Blomer
2. Jonas Hanfeld, Jonas Rembser, Marta Czurylo, Olivier Couet, Philippe Canal **(FNAL)**,
3. Vasil Vassilev **(Princeton)**, Lorenzo Moneta, Monica Dessole, Vincenzo Padulano, Serguei Linev **(GSI)**
4. **New:** Jack Parolini, QUEST-35, QUEST-37, LD-55

Plus students, working with us a few months, up to ~1 year

Not everybody in this slide is 100% dedicated to one project, but most are.

The Team and the Focus Areas



Platform Support

- Windows
- Linux
- Macos

- Build system
- Build Infra
- Training

Team members are encouraged to be involved in more than one ROOT component, as well as to take part to baseline work.

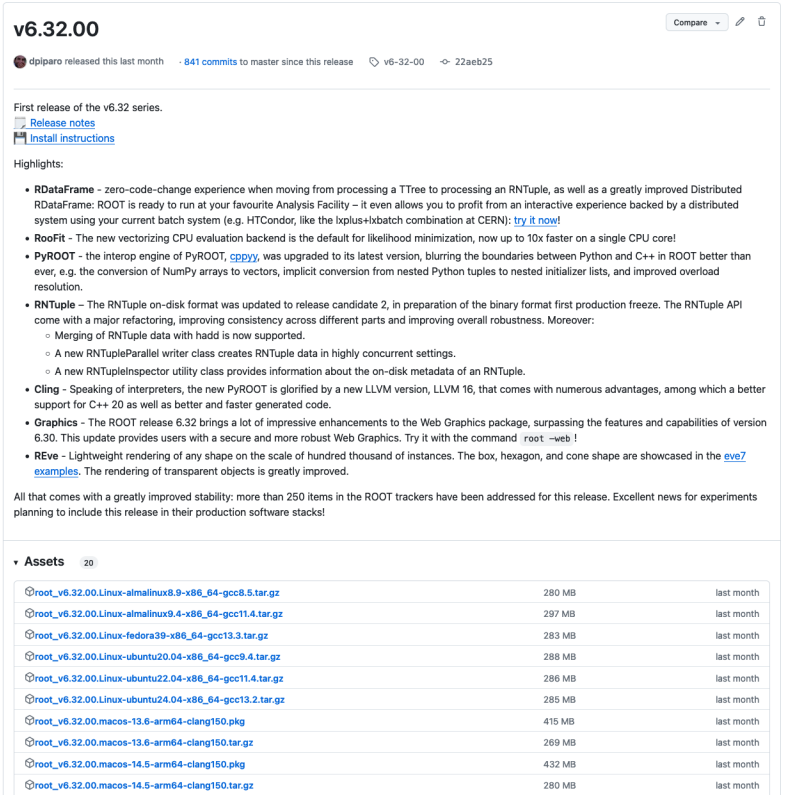
The structure of the project, components and people's focus has been greatly simplified for this slide: to be taken with a grain of salt!

Highlights
(all in production in 6.32!)

Releases during Q2

- 6.32.00 released on May 28th (promised for the 4th week of May)
  - 6.32.02 released last week – minor fixes, concentrated on RooFit, which went through major oprimisations
  - Lesson learned: almost all issues could have been caught with 6.32.00-rc1, we'll make sure Release Candidates are advertised more widely (maybe also providing some binaries/cvmfs installations/Conda package to ease early testing)
- 6.30.08 released last week, upon LHCb request
  - **24h between request of the experiment and delivery of the release**

Timely delivery of high quality releases, according to schedule or upon experiment request.



v6.32.00

Compare

dpiparo released this last month · 841 commits to master since this release · v6-32-00 · 22aeb25

First release of the v6.32 series.
Release notes
Install instructions

Highlights:

- **RDataFrame** - zero-code-change experience when moving from processing a TTree to processing an RNTuple, as well as a greatly improved Distributed RDataFrame: ROOT is ready to run at your favourite Analysis Facility – it even allows you to profit from an interactive experience backed by a distributed system using your current batch system (e.g. HTCondor, like the lxplus+lxbatch combination at CERN): try it now!
- **RooFit** - The new vectorizing CPU evaluation backend is the default for likelihood minimization, now up to 10x faster on a single CPU core!
- **PyROOT** - the interop engine of PyROOT, cppyy, was upgraded to its latest version, blurring the boundaries between Python and C++ in ROOT better than ever, e.g. the conversion of NumPy arrays to vectors, implicit conversion from nested Python tuples to nested initializer lists, and improved overload resolution.
- **RNTuple** – The RNTuple on-disk format was updated to release candidate 2, in preparation of the binary format first production freeze. The RNTuple API come with a major refactoring, improving consistency across different parts and improving overall robustness. Moreover:
  - Merging of RNTuple data with hadd is now supported.
  - A new RNTupleParallel writer class creates RNTuple data in highly concurrent settings.
  - A new RNTupleInspector utility class provides information about the on-disk metadata of an RNTuple.
- **Cling** - Speaking of interpreters, the new PyROOT is glorified by a new LLVM version, LLVM 16, that comes with numerous advantages, among which a better support for C++ 20 as well as better and faster generated code.
- **Graphics** - The ROOT release 6.32 brings a lot of impressive enhancements to the Web Graphics package, surpassing the features and capabilities of version 6.30. This update provides users with a secure and more robust Web Graphics. Try it with the command `root -web` !
- **REve** - Lightweight rendering of any shape on the scale of hundred thousand of instances. The box, hexagon, and cone shape are showcased in the eve7 examples. The rendering of transparent objects is greatly improved.

All that comes with a greatly improved stability: more than 250 items in the ROOT trackers have been addressed for this release. Excellent news for experiments planning to include this release in their production software stacks!

Assets 20

root_v6.32.00.Linux-almalinux8.9-x86_64-gcc8.5.tar.gz	280 MB	last month
root_v6.32.00.Linux-almalinux9.4-x86_64-gcc11.4.tar.gz	297 MB	last month
root_v6.32.00.Linux-fedora39-x86_64-gcc13.3.tar.gz	283 MB	last month
root_v6.32.00.Linux-ubuntu20.04-x86_64-gcc9.4.tar.gz	288 MB	last month
root_v6.32.00.Linux-ubuntu22.04-x86_64-gcc11.4.tar.gz	286 MB	last month
root_v6.32.00.Linux-ubuntu24.04-x86_64-gcc13.2.tar.gz	285 MB	last month
root_v6.32.00.macos-13.6-arm64-clang150.pkg	415 MB	last month
root_v6.32.00.macos-13.6-arm64-clang150.tar.gz	269 MB	last month
root_v6.32.00.macos-14.5-arm64-clang150.pkg	432 MB	last month
root_v6.32.00.macos-14.5-arm64-clang150.tar.gz	280 MB	last month

Proposed Release Schedule for 2024

November W1, development release, short term support:

- Name still undecided: 6.33 or 6.34
- Fit all the features we have at that time
- Branch 1.5 months before: September W3
- Release Candidate 2 weeks before: October W3
- The LLVM version is not yet decided, but the plan and decision point ahead of us are clear:
  - We continue with the endeavour of upgrading ROOT to LLVM 18. Decision point in 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.
  - So far only 10 tests of the entire ROOT battery are failing – debugging them one by one

A clear release plan ahead of us

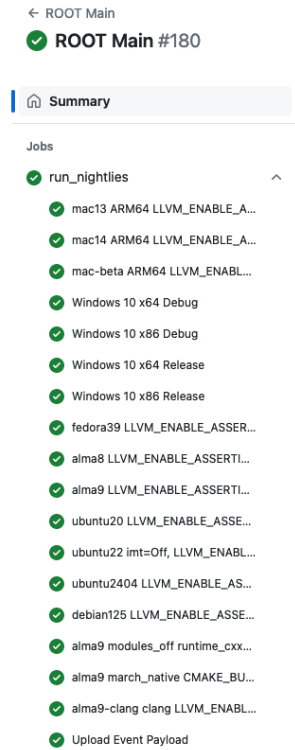
2025 release schedule depends on LHC schedule

- Assuming current schedule: discussion with experiments needed towards the end of 2024

Patch releases will be provided regularly throughout the next months, also according to the needs of experiments

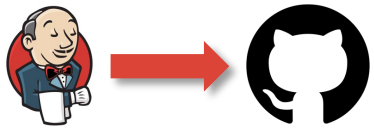
New GitHub-Based CI

- **ROOT CI fully based on Github Actions** (Jenkins kept as backup)
- 5 branches regularly integrated: 6.26, 6.28, 6.30, 6.32, master
- Win/Mac Physical/Virtual hosts
  - Macos versions: 12, 13, 14 and 15 beta – 12 likely to be removed in October (if previous years' schedule respected by Apple)
- Linux: fully containerised setup - generic hosts, building and testing in containers
  - Test many Linux distros with little effort
  - Experiment setups to increase coverage: clang builds, builds w/o cpp modules, march=native compiler flags, asan …
  - Possible thanks to the great support of CERN's IT-CD (Compute & Devices)
- **Run all >2k tests for every platform, every PR and every nightly build**



~ 750 cores!

A modern, robust, and efficient CI that unlocks prolific development!



Python Interface: PyROOT

ROOT has been part of the Python ecosystem since years. Clear direction: reinforce this position, prioritizing ROOT experience for Python users. Example actions taken:

1. **ROOT's C++-Python "interop engine" (cppyy) was updated to its latest version for release 6.32**
   - Faster, less custom patches to maintain (~20), much improved C++ support (e.g. templates)
   - Tested promptly by CMS in the production software stack
   - Made available to all users + ATLAS/LHCb in *dev3 LCG stack on CVMFS* to be tested from day 1
2. For the first time, a demo infrastructure to install ROOT with *pip* was set up
   - Plan: continue to work on the demonstrator, to assess whether it can be used as one of the main ROOT binaries distribution channels – **Try it and give us feedback!**

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

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

Teaching

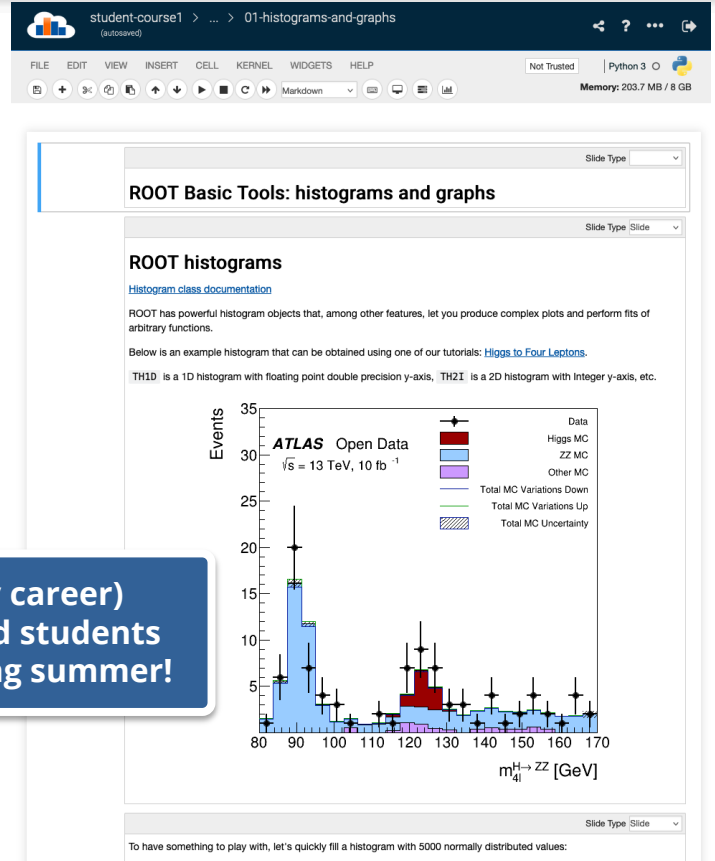
Intense teaching activities ongoing

- 2 IRIS-HEP/HSF Python for Analysis Training
  - June 5th , July 16th
- 4 CERN Summer Student ROOT Workshops
  - June 14th, June 21st, July 10th, July 24th

Main features:

- Enthusiastic and capable attendees
- Fully based on Jupyter notebooks and CERN SWAN
  - Opportunity to expose the new Javascript-based graphics

Re-organization of ROOT's training material ongoing, also based on the feedback collected during those events and the surveys returned by attendees.



>100 (early career) scientists and students reached during summer!

Click here to start the course! Open in SWAN

Conda

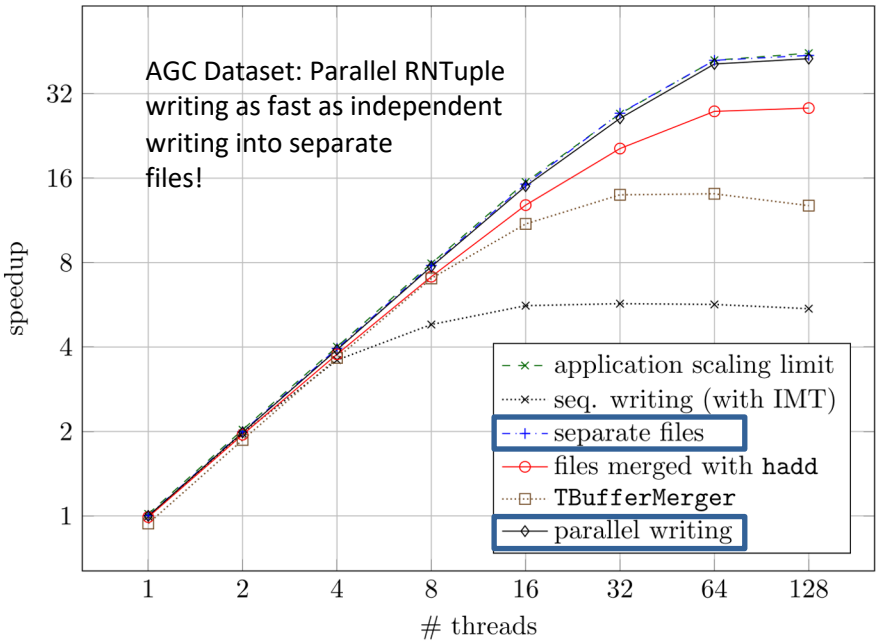
Excellent vector to obtain ROOT, validated by an established industry community

Several platforms tested on Conda infrastructure!

- 6.32.00 available, 6.32.02 imminent
- Interest expressed by CMS (CAT group), LHCb, several questions on the Forum
- Purdue T2 this post: interest to use GPU enabled RooFit usage from a Conda environment
  - Moreabout this later

```
$ conda config --set channel_priority strict
$ conda create -c conda-forge --name <my-environment> root
$ conda activate <my-environment>
```

- **For the first time, complete CMS MiniAOD were stored in RNTuple**
  - Several thousand, complex branches).
  - That's a milestone that took 2 years to reach, in close collaboration with CMS Core SW team!
- **The RNTuple on-disk format updated to RC-2**: milestone towards the binary format freeze.
- The RNTuple API major refactoring, improving consistency across different parts and improving overall robustness.
- Merging of RNTuple data with hadd is now supported.
- **RNTupleParallelwriter class creates RNTuple data in highly concurrent settings.**
- A new RNTupleInspector utility class provides information about the on-disk metadata of an RNTuple.



- The new vectorizing CPU evaluation backend is the default for likelihood minimization
- In general, up to **10x faster on a single CPU core**
- **Straightforward to GPU support**, even against an already existing ROOT installation (plays well with Conda, proposed to Purdue sys admins)

RDataFrame and Analysis Facilities

RDF runs on distributed systems, already today: DistRDF

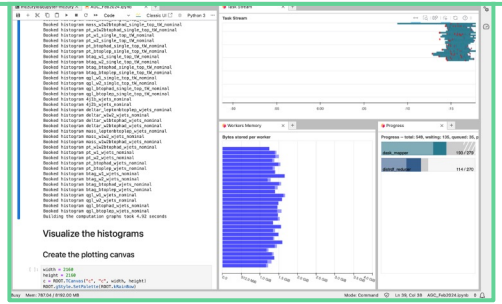
- Same analysis code of local execution
- **Shell and notebooks**
- Runs everywhere, e.g.
  - Lxplus @CERN
  - Bare-metal clusters
  - HPC environments
  - **Analysis Facilities!**

See linked papers for more information

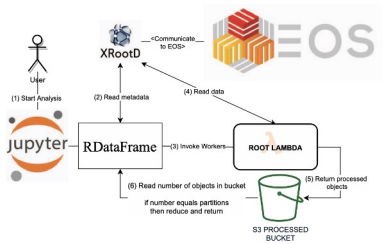
ROOT is ready to run at your favourite analysis facility, today

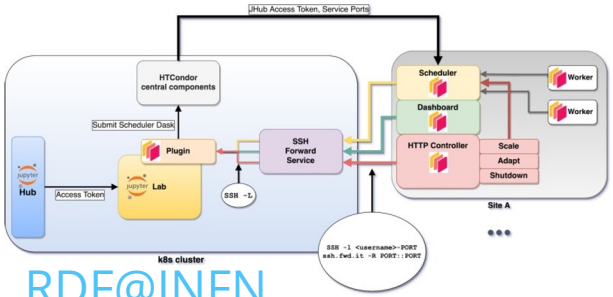
RDF+RNTuple@SWAN

```
"root://eospublic.cern.ch//eos/root-eos/AGC/"
"root://eospublic.cern.ch//eos/root-eos/AGC/rntuple/"
```

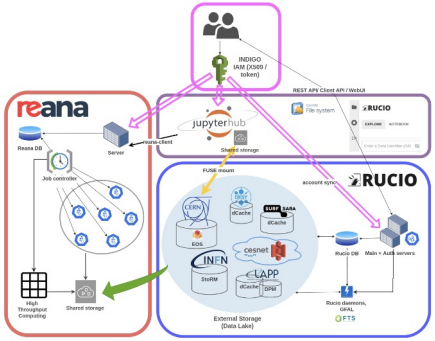


RDF@AWS





RDF@INFN

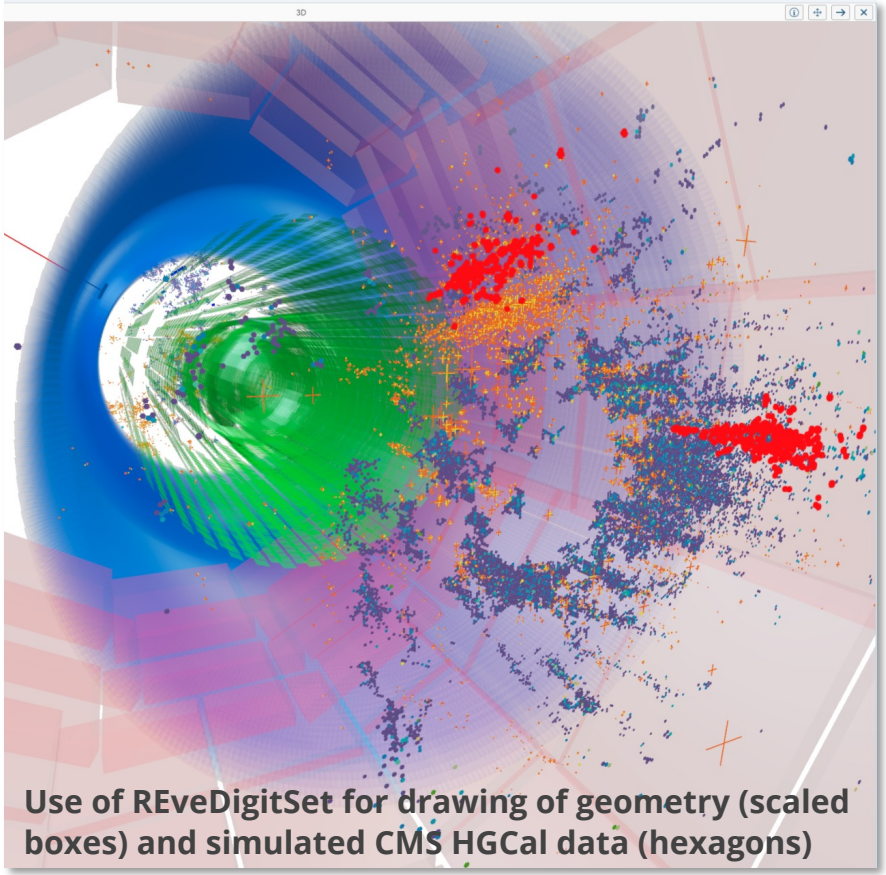


RDF@VRE

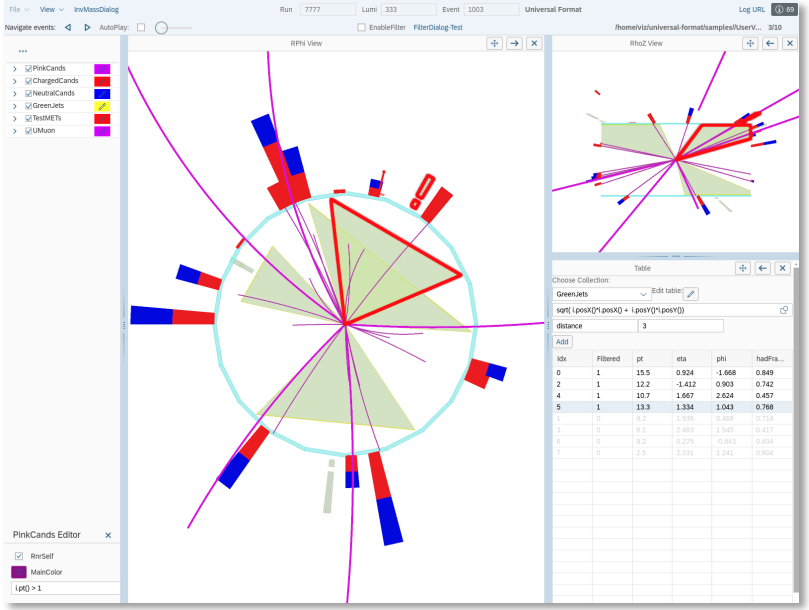
REve: A Unique Visualisation Framework

- Based on the new ROOT web-based graphics
  - Server-client architecture, has a client-only JS based version (reduced functionality)
- Direct interaction with original data - REve is a server-client event display
  - Create visual data at runtime, using native EDM: decrease/increase thresholds without an increase of load-time
- True 2D projections for physics analysis
  - RhoZ projection, RPhi view on top of XZ/YZ side-views
  - All projections support fisheye distortion (vertex view) and step-wise scaling (tracker -- calos -- muon system).
  - As these are really 2D projections, not side views, objects can be stacked in a meaningful order: geometry outlines in the back, jets/cals in the middle, tracks in the front
- Instancing of shapes with various degrees of “sameness”
  - Can easily render up to O(100k) or even O(1M) instances
  - Each instance can be independently positioned, scaled, rotated, or coluored (only parameters that differ need to be set)
  - Picking and highlighting of individual instances in a set is supported
  - Tooltip for selected instance can be generated on the server, based on additional information that is available there.
- Based on TEve and almost 2 decades of experience: many features have been created for the needs of ALICE and CMS

REve



Use of REveDigitSet for drawing of geometry (scaled boxes) and simulated CMS HGCal data (hexagons)



Visually engaging, clear and complete interface to a powerful visualisation

Try the JS only version here in your browser!

Conferences

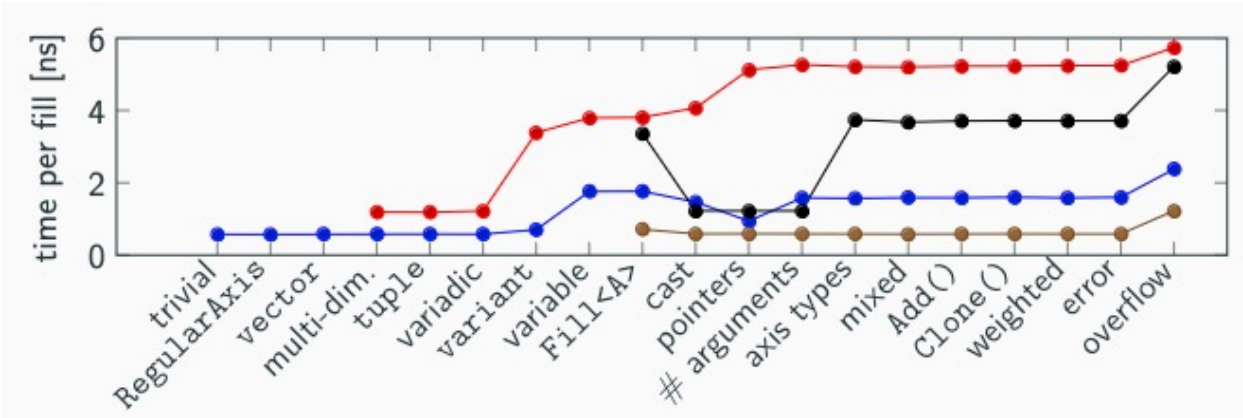
- Two contributions at the WLCG/HSF Workshop in DESY (May)
- 9 oral contributions accepted for CHEP, 2 of them plenary
  - On top of 4 ACAT contributions earlier this year!
- 1 contribution to ICHEP about AD
- 1 contribution at Europar about parallel writing

EURO-PAR
CONFERENCE 2024

ICHEP 2024 | PRAGUE

Path to ROOT 7

- The PTR7 series of meetings started
- For the moment, dedicated to establishing a common understanding of the next release cycle within the team
- Gradually, we plan to include (power) users in the process
- Progress in the design and implementation of "thread-safe histograms": e.g., see this talk

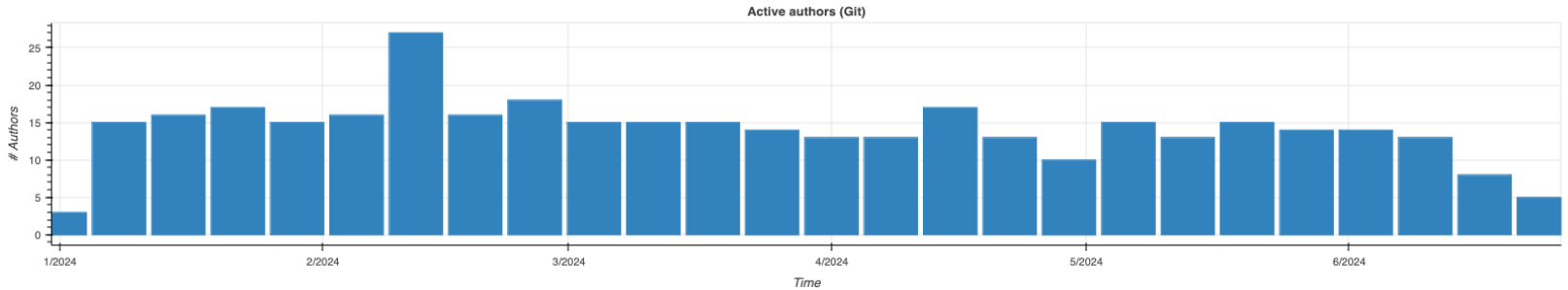


Performance aware design embedded from the start in the process

Q2 Metrics

ROOT Community and Development

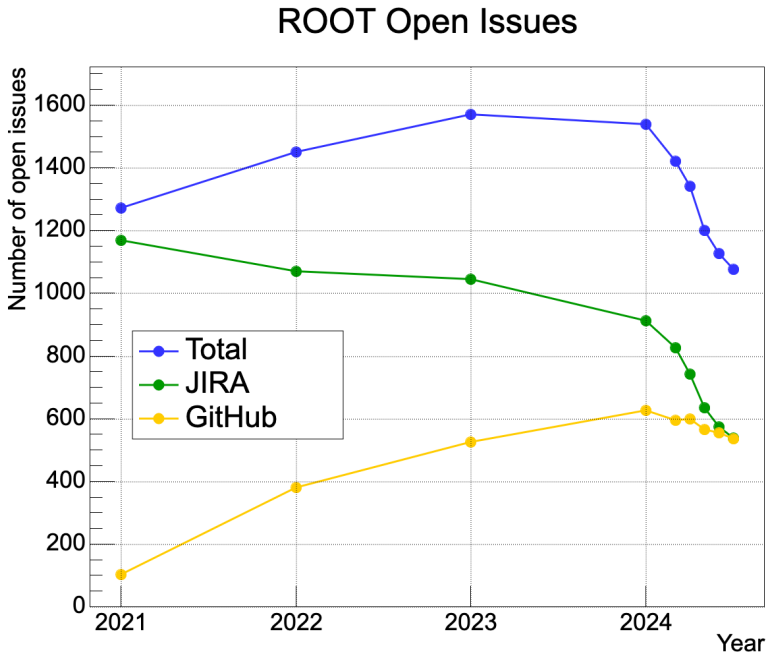
- **ROOT is an open source project, supported by a lively community**
- ROOT Forum: 7h on average to obtain a first response (20h in 2023) - 5.4k posts so far in 2024 (11.8k in 2023)
- **About 15 active developers/contributors at any point in time** (same level of 2023)
- **1301/1284 PR opened/closed, 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*
- 324 new issues created, of which 200 closed
- **785 closed in total in 2024**
- **30% reduction of # issues during H1**

	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
Apr-24	635	566	1201	
May-24	573	555	1128	6.32.00 released
Jun-24	537	539	1076	1 JIRA issue migrated to GH



2024 PoW Completion

DONE
PARTIALLY DONE
NOT DONE

			Priority	Completion Status: 0, .5 or 1		
Builds	1	pip install ROOT for some selected platforms	1	0.5		
and	2	Complete transition to GH Actions, adding GPU runners	1	0.5		
Binaries	3	Reduce number of services hosted by root.cern with a combination of CERN IT central services	1	0		
	4	Win: Replace Debug builds with ReleaseWithDebInfo in the CI	1	0		
	5	Optimise dictionary dependencies to minimise build real time	2	0		
	6	Win: Add support for Ninja	2	1	33.3	%
I/O	1	Support std::variant, both in TTree and RNTuple (CMS)	2	0		
and	2	Support writing objects larger than 1GB (TBufferFile > 1 GB, ALICE)	1	0		
TTree	3	Complete schema evolution improvements	2	0		
	4	Ensure consistency of std::int types across ROOT I/O	2	0		
	5	Address residual scaling issues with MT writing	2	0.5	10.0	%
RNTuple	1	Complete implementation of merging	1	0.5		
	2	Complete implementation of datasets chains	1	0.5		
	3	Limit testing in collaboration with CERN IT	1	0.5		
	4	Follow-up on API review by HEP-CCE	1	0.5		
	5	Implement unsplit ("blobbified") encoding	1	1		
	6	Support for unaligned friends and joins	1	0		
	7	(E) RNTuple: schema evolution	1	0		
	8	Further develop support for lossy compression with low-precision floats	2	0		
	9	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	31.8	%
RooFit	1	Workshop with Experiments: promote features, gather input, speedup integration of RooFit in the existing sw setups	1	1		
	2	Numeric integrals in n-dim with CUDA	1	0		
	3	Evaluation of custom user functions in CUDA	1	0		
	4	Group similar PDFs to speed up evaluation	1	0		
	5	Make the new vectorized CPU likelihood evaluation interface the default	1	1		
	6	Reduce JITting time for AD in RooFit	1	1		
	7	PyROOT: express RooStats configuration with C++-oriented Set* as kwargs	2	0		
	8	Integration of Fumili in RooFit	2	0	37.5	%
RDataFrame	1	Put existing bulk processing in prod	1	0		
	2	DistRDF: reduce memory usage on HTCondor Workers	1	0		
	3	DistRDF: improve user experience when integrated with notebooks and nb services like SWAN	1	0.25		
	4	Make the TTree → RNTuple transition transparent for analysers	1	1		
	5	Further Pythonise the interface	2	0		
	6	Deliver varied snapshots	2	0.25	25.0	%
Math	1	PyROOT: better histos ang graph interoperability with NumPy and UHI protocol	1	0		
	2	Histos: advance current RHist implementation to one testable by experiments	1	0		
	3	Add interface to pass initial error values/cov matrix to Minuit2	1	0		
	4	Release a library for Lorentz vector computations on accelerators in SYCL	1	0.5		
	5	Deliver plan and prototype of algorithmic improvements when dealing with param constraints in ROOT's minimisers	2	0		
	6	PyROOT: Pythonise TF{1,2,3} and numerical algorithms interfaces (e.g. minimisers)	2	0		
	8	Histograms: Model and prototype of pipelining GPU histogram filling	2	0.5	14.3	%
ML/AI	1	Put RBatchGenerator in production	1	0		
	2	Consolidate RBDT	1	1		
	3	Support of integration of SOFIE in experiments Fast Simulation pipelines	1	0		
	4	Add support in SOFIE for NVidia GPUs in CUDA	1	0		
	5	Continue to add support for the ONNX operators requested by experiments	1	0.5		
	6	Make HLS4ML interoperable with SOFIE	2	0		
	7	Streamline ROOT's inference interface, making it able to use models for Python ML frameworks (e.g. Keras/TF) directly	2	0	21.43	%
Visualisation	1	Automated placement/tune of plot elements, "Auto Style"	1	0		
and	2	Add missing features of classic graphics to the web-based one	1	0.5		
UI	3	Automate web-based graphics test suite	1	0		
	4	Add residual missing TEve features to REve, e.g. digit visualisation and text elements overlay	1	0.5		
	5	Visualization of flat ntuples using predefined visual summary data structures	1	0.5		
	6	Improve REve window manager and browser, polish render engine	2	0	25.0	%

Interpreters	1	Cling: identify potential Cling codebase reductions through the reuse of parts of clang-repl	1	0.5		
	2	Cling: cppyy rebase on top of cling/clang-repl	1	0		
	3	Migrate PyROOT to the latest Cppyy	1	1		
	4	Cling: Prototype SYCL support	2	0.5	50.0	%
Doc and	1	Re-evaluate, update, and improve course material, making it more visible and better organised on the website	1	0		
education	2	(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	1	Copyless reading in RNTuple - ALICE	1	1		
	2	Physics objects representations out of NanoAOD in RDataFrame - CMS CAT	1	0		
	3	Bulk Processing + GPU offloading for distRDF - CMS CAT	1	0		
	4	include the open source Tex Gyre Heros clone of Helvetica in root fonts - CMS CAT	2	1		
	5	Multithreading-friendly interfaces to the histogram types - CMS CAT/TSG	1	0		
	6	A library of matrix operations that can run on GPUs - CMS TSG	1	0	33.3	%
				overall:	27.4	%
				overall + Extra Items:	27.9	%

- A large PoW: currently 67 items
- **Current person power in the ROOT team insufficient to deliver all items**
- **New arrivals foreseen during H2**
- External help, i.e. ROOT community (e.g. experiments, but not only) can make the difference, too

PoW is 27.9 % complete if accounting for extra items

For comparison, 2023: 55 items, 55.4%
All details at cern.ch/root-pow

Conclusions

Conclusions

- **ROOT is at the heart of the (HL-)LHC project, fully committed to support its users in the very long term**
  - We are doing our best to concile very long term support and innovation
  - Solid plan ahead, formed listening to all stakeholders
- **2024 PoW 28% complete, issues backlog reduced by 30% since Jan 1st**
- **Extremely active community and team during Q2 (and Q1)**
  - **~1300 PR opened – the Open Source model pays off**
  - 7h to get a first answer on the ROOT Forum
  - Strong presence at trainings (~10) and conferences (~20 contributions during 2024)
- Major release and few patch **releases delivered on time or with minimal latency upon experiment request**. Highlights:
  - 10x faster RooFit, AF support, new PyROOT, Conda, prototype Pip-install-ROOT, Event display, new powerful CI