ROOT Plan Of Work 2024

https://root.cern/

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

15-01-2024

This Talk

The first ROOT PoW talk, very early in the year.

Some feedback from experiments, ‘godparents’ as well as users has been already incorporated. More is welcome, and sought after.

The PoW will also be presented to godparents (on January 23) and experiments (on January 24).

This talk is divided three parts:

1. Some 2023 statistics, and how to use the various indicators during 2024
2. Short summary of the achievements in 2023
3. Plan of work 2024, divided by focus area

Project Leader Change

- After 6 years as ROOT Project Leader, Axel Naumann was asked and decided to take over an important coordination role in the Next Generation Triggers project at CERN
- As of January 1st, **D. P. replaces Axel as ROOT Project Leader; the change had the support of Axel and the team**
- It was a pleasure and honour to work with you, Axel. We will miss your competence, kindness and creativity - thanks for everything you have done for the ROOT Project and Team as a user, developer and leader.

The Strategic Goal of ROOT

Our commitment to the (HL-)LHC project as well as present and future HE(N)P computing at large does not change

Our goal is to 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

1. Some 2023 Statistics

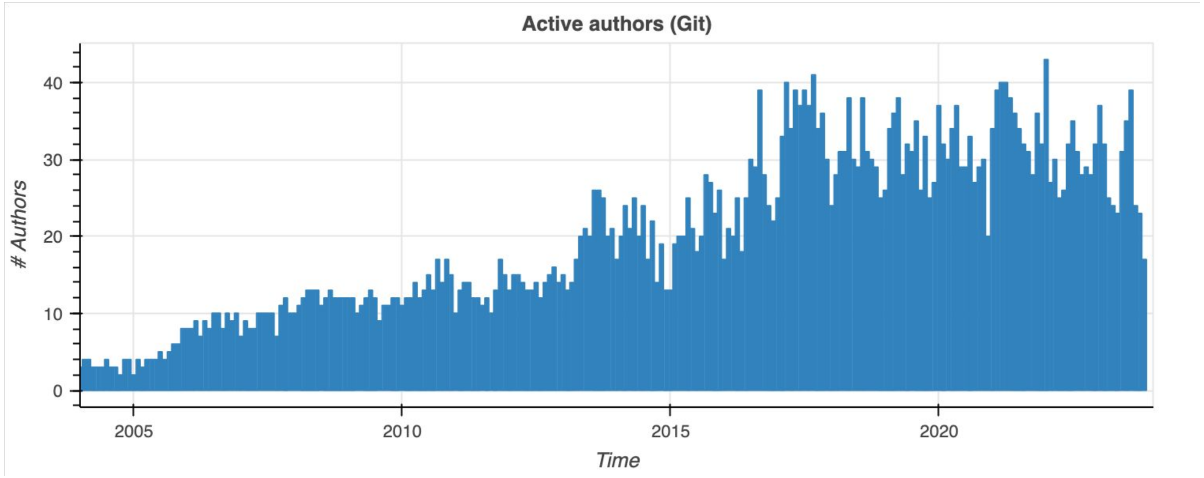
The code related statistics were obtained with Cauldron for the ROOT main repository only

Code Changes and Code Reviews

- 1740/1698 Pull Requests opened/closed (1900/1891 in `22)
- Substantial effort invested in code reviews: **only 1 day (median) needed to close a PR!**

A healthy codebase, development supported by *diverse* authors

(see also the Team composition in the backup)



Issues in 2023 (and 2022, 2021)

Not all issues are bugs, e.g. feature requests.

Work needed to address an issue varies.

- **435/441 created/closed** (`22 520/360, `21 650/480)
- Current open issues: 1539
  - 627+912 on GitHub+JIRA (`22 525+1045, `21 380+1071)
  - Issue tracker moved to GH, JIRA cannot be used for new issues
- Median time to close new issues was 14 days (`22 12, `21 10)

The backlog still there, even if slightly reduced during 2023

Challenge for 2024: while keeping high quality support level, increase development capacity and reduce backlog (see 2024 team composition and size later in the slides)

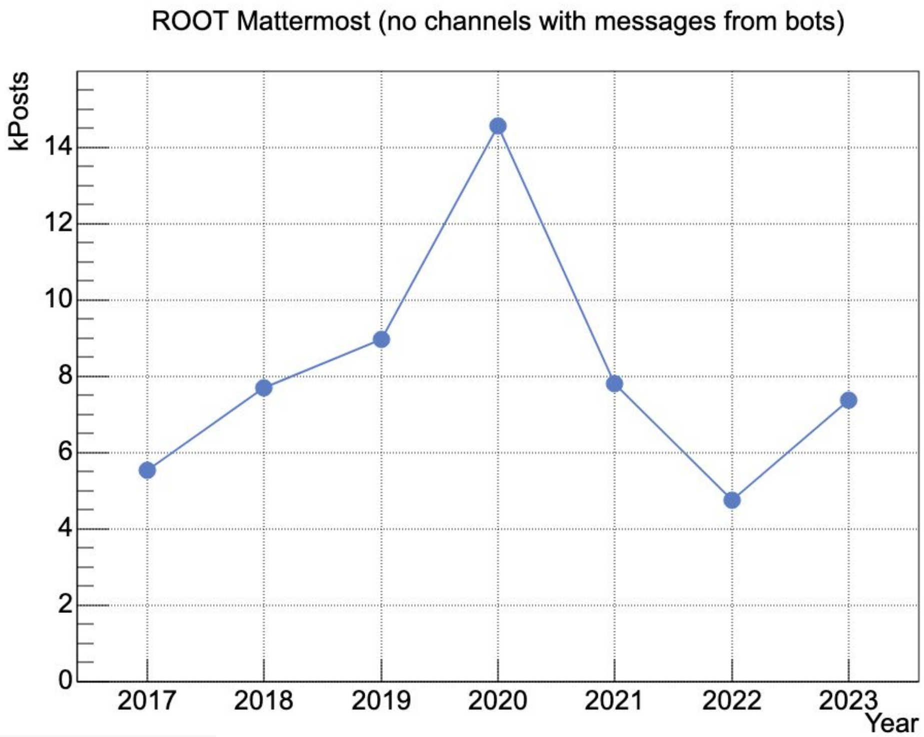
Support during 2023

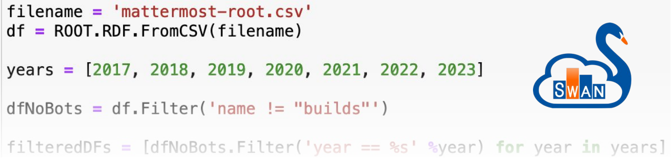
- **Main support channel: the ROOT forum, CERN's #1 Forum**
  - **11.8k posts, 1.2k registrations**
  - 20h on average to obtain the 1st response
  - Answers don't come only from the ROOT team
  - Useful tool to understand what is being picked up / where improvements should be
- Expert (but not only) exchanges also happen on the ROOT Mattermost team
- In addition to the Forum/Mattermost/GitHub, we have direct communication with experiments (e.g. core sw and analysis tools teams)

Lively community and clearly defined communication channels

Thanks to Ismael Posada, CERN IT, for comparing to all CERN Discourse Fora!

Mattermost Stats


ROOT Mattermost (no channels with messages from bots)


```
filename = 'mattermost-root.csv'
df = ROOT.RDF.FromCSV(filename)

years = [2017, 2018, 2019, 2020, 2021, 2022, 2023]

dfNoBots = df.Filter('name != "builds"')

filteredDFs = [dfNoBots.Filter('year == %s' %year) for year in years]
```

Thanks to Adrian Mönnich, CERN IT, for extracting the input data!

Some Quantities to Keep an Eye On

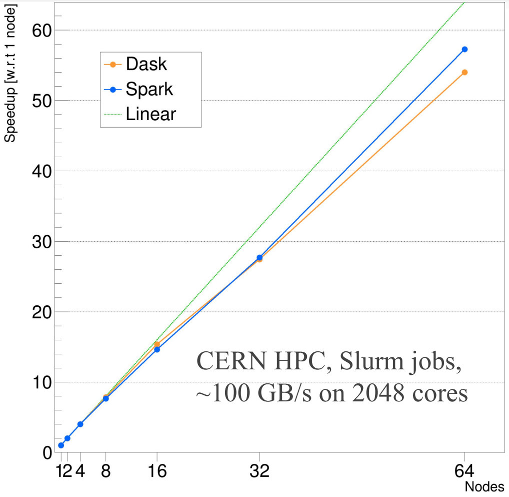
During 2024, we'll keep track of the following quantities and, if needed, adjust the way we operate accordingly:

- Forum: average time to provide an answer, # posts, # registrations
- Issue tracker: # issues closed (also normalised with the # issues opened), backlog length
- Code changes: # PRs, #days to provide a review

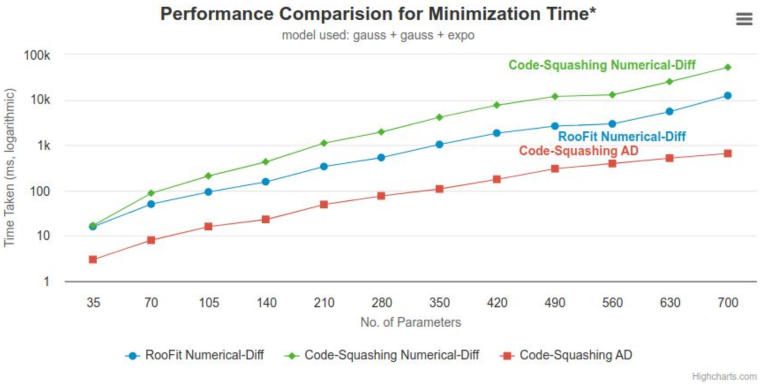
2. 2023 Highlights

Analysis, ML and RooFit

- Demonstrated fitness of RDataFrame for Analysis Facility model @INFN and @CERN with SWAN (paper)
- **DistRDF scalability on thousands of cores** @ HPC centres (JGC publication, Julich)
- Contributing the Analysis Grand Challenge candle (runs with RDF in <1 minute, CHEP23)
- User experience: progress bar, live visualizations
- Extend SOFIE* features (operators and GNN)
- Batch Generator for ML training
- Add support for Automatic Differentiation in RooFit



CERN HPC, Slurm jobs, ~100 GB/s on 2048 cores

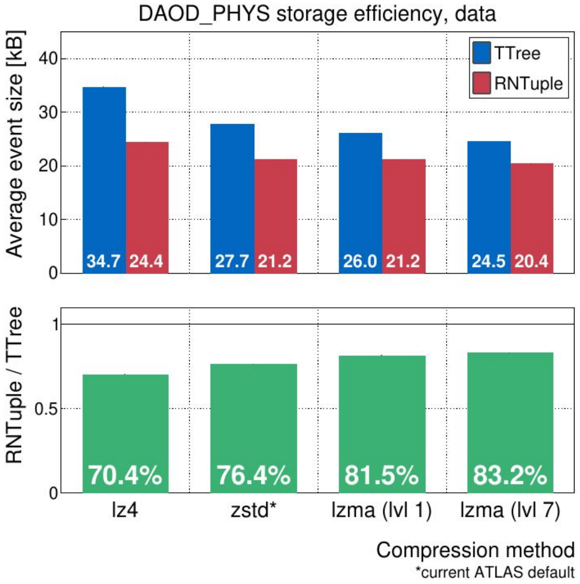


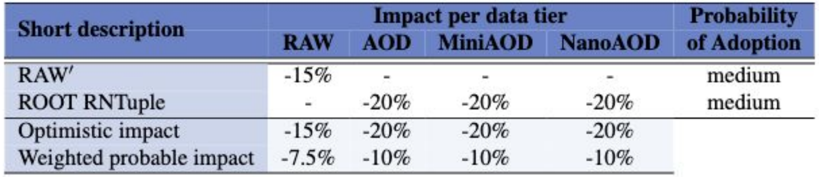
** DistRDF: RDataFrame running on many nodes, in a distributed manner.*

*** SOFIE (System for Optimized Fast Inference code Emit): generates C++ functions invokable for the inference of trained neural network models. It takes ONNX model files as inputs and produces C++ header files ready to be compiled or interpreted.*

RNTuple

- The RNTuple Workshop in November
- **Demonstrated up to 20% smaller dataset size and 2.8x faster time-to-result for analysis**
- Support for LHCb analysis ntuples, CMS Nano and…
- … full ATLAS EDM (but RAW, not written in TTree format): achieved also thanks to the efforts of the ATLAS Core SW team (thanks!)
- End user experience: RNTuple Importer & Inspector
- Late model extension, I/O read rules, custom collections fields, std::(unordered_)set, std::(unordered_)map fields



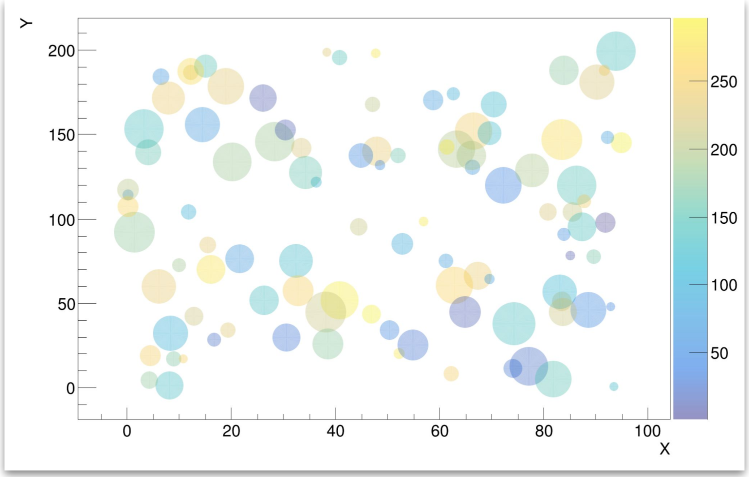


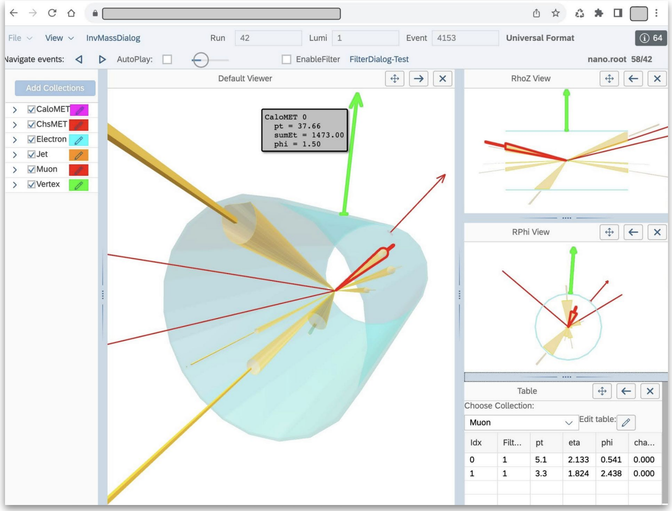
Short description	Impact per data tier				Probability
	RAW	AOD	MiniAOD	NanoAOD	of Adoption
RAW$'$	-15%	-	-	-	medium
ROOT RNTuple	-	-20%	-20%	-20%	medium
Optimistic impact	-15%	-20%	-20%	-20%	
Weighted probable impact	-7.5%	-10%	-10%	-10%	

CMS expectations for Run 4, 10-20% reduction, honoured - CMS-NOTE-2022-008

Graphics and Visualisation

- Web and classic graphics have almost the same functionality (see JSROOT 7.5.0 RN)
- First implementation of GPU-powered Render Core (see paper), overlays and fonts, digit visualization
- TScatter: 4 variables in the same plot (x,y,colour,marker size)
- Preparatory work for the *auto style*: completely automatic primitive placement





Run 42 Lumi 1 Event 4153

Navigate events: AutoPlay: □ □ EnableFilter FilterDialog-Test nano.root 58/42

☑ CaloMET
☑ ChsMET
☑ Electron
☑ Jet
☑ Muon
☑ Vertex

```
CaloMET 0
  pt = 37.66
  sumEt = 1473.00
  phi = 1.50
```

Choose Collection: Muon

Edit table:

Idx	Filt...	pt	eta	phi	cha...
0	1	5.1	2.133	0.541	0.000
1	1	3.3	1.824	2.438	0.000

Interpreters, Builds and Infrastructure

Interpreters

- **Upgrade to LLVM 16 from LLVM 13, achieving increased sustainability**
  - Meticulous planning, changes merged into ROOT, *zero* tests broken
  - Increased automation of the LLVM upgrade procedure put in place
  - 86 patches in ROOT-llvm13 → 66 patches for ROOT-llvm16 (10 are backports from upstream)
- Speedup *import ROOT* by factor 5x
- Faster (10x) and not-memory-hogging Python value printing

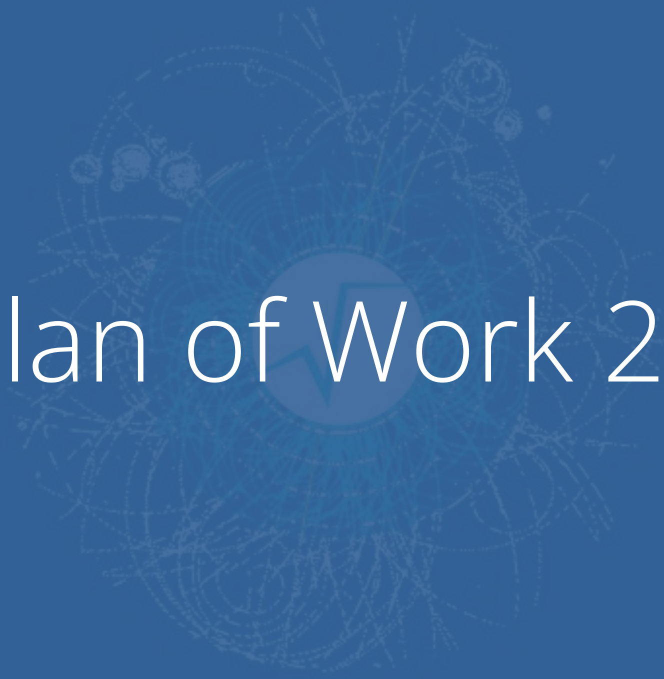
Builds and infrastructure:

- New CI based on GitHub actions
- Windows debug assertion builds (which led to the fix of numerous subtle issues → ROOT is now more robust)

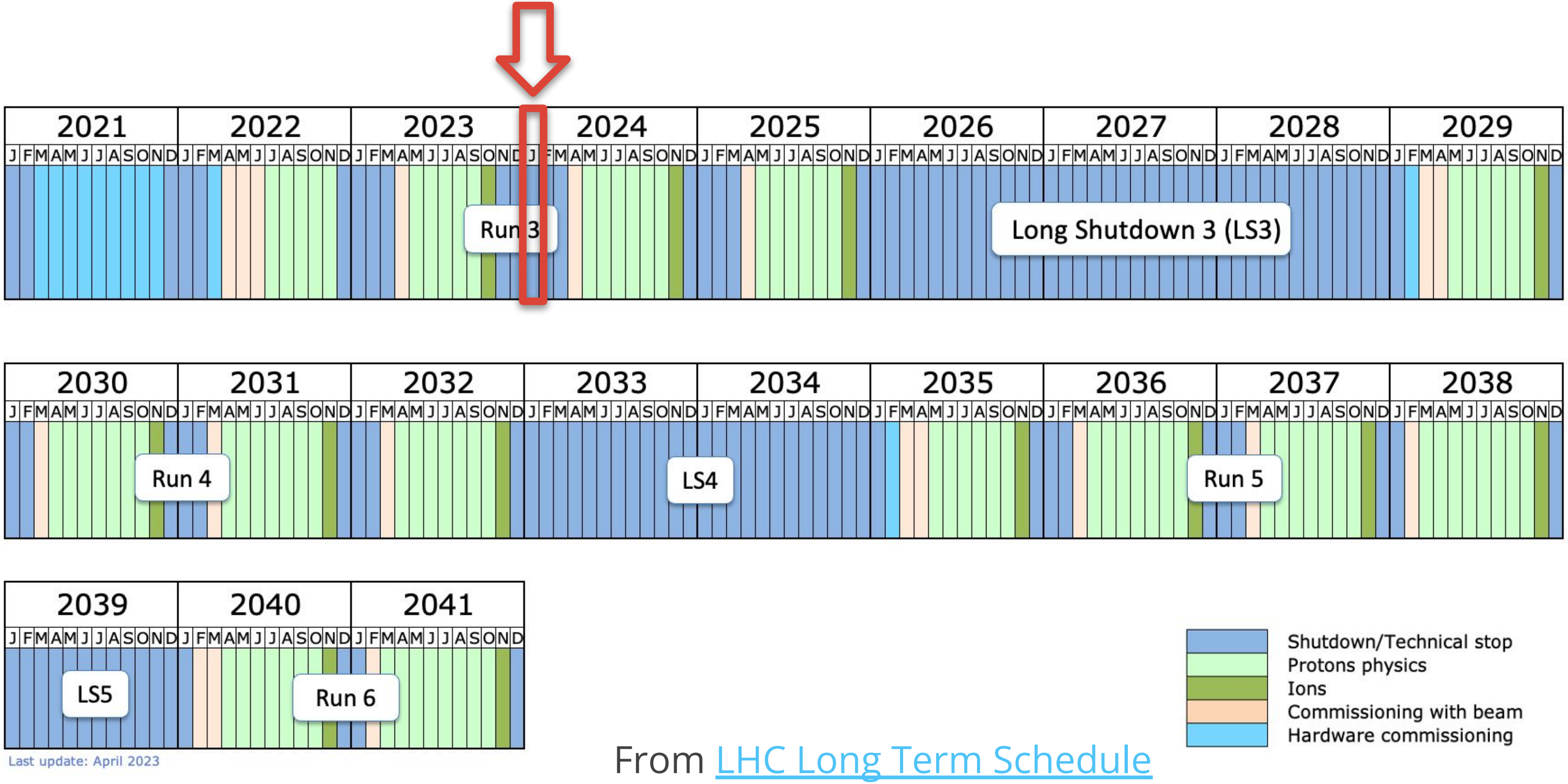
Presentations, Publications and Trainings

- Substantial presence at CHEP: 11 contributions
  - See 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11
- Numerous talks at experiments' meetings and O&C weeks
- **Peer-reviewed publications**
  - *Prototyping a ROOT-based distributed analysis workflow for HL-LHC: the CMS use case* 10.1016/j.cpc.2023.108965
  - *Leveraging state-of-the-art engines for large-scale data analysis in High Energy Physics* 10.1007/s10723-023-09645-2
  - *Leveraging an open source serverless framework for high energy physics computing* 10.1007/s11227-022-05016-y
- 2 Python for Analysis Trainings, 3 instances of CERN Summer Student ROOT Training

3. Plan of Work 2024



LHC Schedule



From [LHC Long Term Schedule]

ROOT Team in 2024

- **ROOT is an international collaboration**, with steady/formal contribution from CERN, FNAL, GSI, Princeton, and UCSD and a *veritable nebula* of other contributors
- Changes in the core team:
  - **One senior developer captured back, one left**
  - ***Two* long-term positions** (5+3 years) to start in 2024
  - One CERN Staff, two Graduate and one Student positions to be opened in 2024
- Significant influx of young contributors, supported by several channels (e.g. EU projects, experiments, CERN EP R&D, NSF)
- 2023 team size limited by available supervision capacity
  - Will be increased in 2024, if team composition is preserved

Opportunity to redistribute load of operations, development and research

Effort and Priorities

- All items have a **priority level: 1 and 2. The higher is 1**
  - Stretch goals are summarised in the backup
- The Initials of the team members working on the items are on the slides
  - The legend can be found in the backup
  - Not all team members dedicate 100% of their time to ROOT
- In all areas **the main priority continues to be the support of the running experiments to address urgent issues**

I/O

PC, JB, JH, VP, FdG

Priority 1

- Support `std::variant`, both in `TTree` and `RNTuple` (CMS)
- Support writing objects larger than 1GB (TBufferFile > 1GB, ALICE)

Priority 2

- Complete the schema evolution improvements
- Ensure consistency of `std::int` types across ROOT I/O
- Address residual scaling issues with multithreaded writing

RNTuple 1/2

PC, JB, JH, VP, FdG

Priority 1:

- Support for chains of datasets and merging
- Complete implementation of schema evolution based on existing I/O customisation support
- Limit testing and validation in collaboration with CERN IT, which is helping with this effort (thank you!): AGC, experiments workflows, mass storage
- Follow-up on API review by HEP-CCE during the first half of the year (thanks US ATLAS and US CMS!)

Features, API and testing

RNTuple 2/2

PC, JB, JH, VP, FdG

Priority 2:

- Support for unaligned datasets join (aka unaligned friends)
- Further develop support for lossy compression with low-precision floats
- Design compression dictionaries and understand implications for specification
- Implement unsplit-encoding, i.e. objects streamed as blobs and not N columns, one per class data member (important to store e.g. TH1s)
- First implementation of highly scalable parallel writing
- Organise a Design Workshop to discuss intra-link events, metadata, native SoA layout for events

Builds and Infrastructure

BB, DP, VP, JR

Priority 1:

- `pip install ROOT` for some selected platforms
- Complete transition to GH Actions, adding GPU runners
- Reduce the number of services hosted by the root.cern server with a combination CERN IT central services
- Win: Replace Debug builds with ReleaseWithDebInfo in the CI

Priority 2:

- Optimise dictionary dependencies to minimise build real time
- Win: Add support for Ninja

Consolidation and sustainability

RooFit

LM, JR, VV

Priority 1:

- Workshop with experiments: promote new RooFit features, gather input, speedup integration of RooFit in the existing software setups for statistical interpretation
- Numeric integrals in n-dim in Cuda
- Evaluation of custom user functions in Cuda
- Make the new vectorized CPU interface the default
- Group similar PDFs to speed up evaluation
- Reduce JITting time for automatic Differentiation in RooFit

Runtime performance and heterogeneity

Priority 2:

- PyROOT: express RooStats configurations via C++-oriented Set* as kwargs
- Integration of Fumili in RooFit

Graphics and Event Display

MT, AM, SL, OC

Priority 1:

- Automated placement/tune of plot elements, “Auto Style”
- Add missing features of classic graphics to the web-based one
- Automate web-based graphics test suite
- Add residual missing TEve features to REve, e.g. digit visualisation and text elements overlay
- Visualization of flat ntuples using predefined visual summary data structures

Priority 2:

- Improve REve window manager and browser, polish render engine

Consolidation and sustainability

Analysis with RDataFrame

MC, VP, DP

Priority 1:

- Put existing bulk processing feature in production
- DistRDF: reduce memory usage on HTCondor Workers
- DistRDF: improve user experience when integrated with notebooks and notebook services, e.g. SWAN
- Make the TTree → RNTuple transition transparent for analysers

Priority 2:

- Further pythonise interface
- Deliver varied snapshots

Runtime performance and ergonomic interfaces

Release Schedule

- Usual date for production release (“even number”): November
  - This year it would be 6.32.00
  - If needed the release can be anticipated
  - The foreseen feature set will have to be reduced accordingly
  - Some periods are more problematic than others, e.g. many team members will attend CHEP, or will be on holiday in July/August
- Usual date for development release (“odd number”): May
  - Not released in a while
  - The master branch has many new features which could be beneficial to many, if exposed
  - Discussions ongoing in the team about the opportunity to release in May

Interpreters

VV, DS

Priority 1:

- Cling: identify potential Cling codebase reductions through the reuse of parts of clang-repl
- Cling: cppyy rebase on top of cling/clang-repl
- Migrate PyROOT to the latest Cppyy

Priority 2:

- Cling: Prototype SYCL support

Sustainability, heterogeneity and functionality

Math

LM, JR, MD, JH, DS

Priority 1:

- PyROOT: better histos and graphs interoperability with numpy arrays, honour UHI protocol
- Histograms: advance current RHist implementation to one testable by experiments
- Improve interface to pass initial error values or covariance matrix to Minuit2
- Release a library for Lorentz vector computations on accelerators in SYCL (also using generic n-dim arrays as inputs)

Priority 2:

- Deliver plan and prototype of algorithmic improvements when dealing with parameter constraints in ROOT’s minimisers
- PyROOT: Pythonise TF{1,2,3} and numerical algorithms interfaces (e.g. minimisers)
- Prototype SYCL kernels to be JITted (see Interpreters objectives)
- Histograms: Model and prototype of pipelining GPU histogram filling

Machine Learning

LM, JR

See Lorenzo's talk Vision for a new ML/AI activity !

Priority 1:

- Put RBatchGenerator in production
- Consolidate RBDT
- Support of integration of SOFIE in experiments Fast Simulation pipelines
- Add support in SOFIE for NVidia GPUs in CUDA
- Continue to add support for the ONNX operators requested by experiments

Priority 2:

- Make HLS4ML interoperable with SOFIE
- Streamline ROOT's inference interface, making it able to use models for Python ML frameworks (e.g. Keras/TF) directly

We want to support experiments inference (C++) for cases that are difficult to implement or require heavy dependencies.

We don't want to compete with existing industry tools for training.

Documentation and Education

MC, VP, OC, DP et. al

Priority 1:

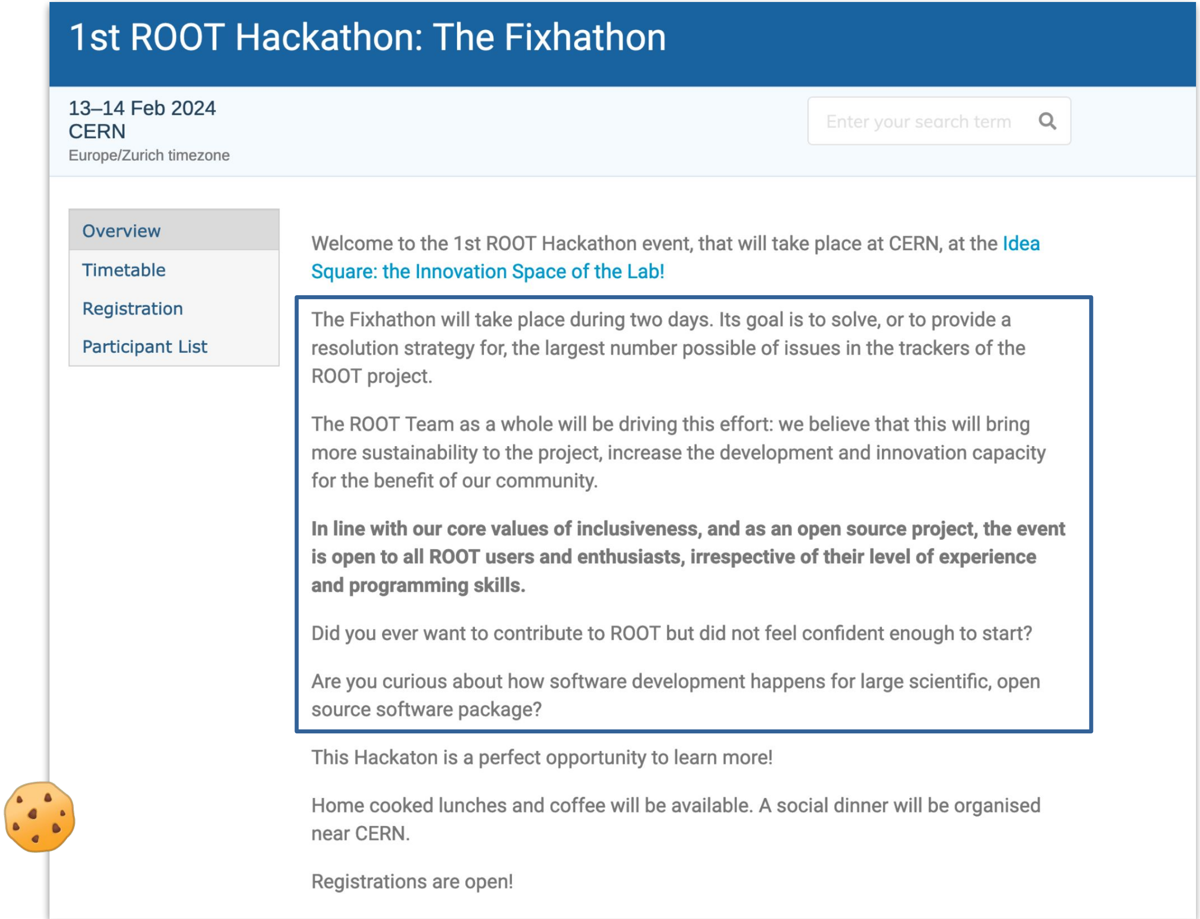
- Re-evaluate, update, and improve course material, making it more visible and better organised on the website
- (Re-)evaluate tutorials, eliminating what’s outdated, what newer features would benefit from a (better) tutorial, improve visibility on the website

Clarity and discoverability

And Two Additional Small Innovations

- We’ll start the *Path to ROOT 7 (PTR7)* process internally in the team
  - Well scoped, result oriented blue printing discussions about ROOT’s 7th release cycle
  - Goal: reach consensus on what should be the content of the ROOT 7.00.00 release
  - The natural continuation of the discussions about ROOT 7 components already proposed within the 6th Cycle, such as RDataFrame and RNTuple
- The first “ROOT Hackathon: *The Fixathon*” event will take place, attended by the ROOT team and open to everybody
  - CERN Idea Square, February 13th and 14th, whole day
  - See next slide!

Registrations are open!



1st ROOT Hackathon: The Fixhathon

13–14 Feb 2024
CERN
Europe/Zurich timezone

Welcome to the 1st ROOT Hackathon event, that will take place at CERN, at the Idea Square: the Innovation Space of the Lab!

The Fixhathon will take place during two days. Its goal is to solve, or to provide a resolution strategy for, the largest number possible of issues in the trackers of the ROOT project.

The ROOT Team as a whole will be driving this effort: we believe that this will bring more sustainability to the project, increase the development and innovation capacity for the benefit of our community.

In line with our core values of inclusiveness, and as an open source project, the event is open to all ROOT users and enthusiasts, irrespective of their level of experience and programming skills.

Did you ever want to contribute to ROOT but did not feel confident enough to start?

Are you curious about how software development happens for large scientific, open source software package?

This Hackaton is a perfect opportunity to learn more!

Home cooked lunches and coffee will be available. A social dinner will be organised near CERN.

Registrations are open!

🍜 ☕ 🍪

© CERN BUL-NA-2015-100

https://indico.cern.ch/e/1stROOTHackaton

Wrap up

Wrap-Up

- Our community, made of users and experiments, is our strength
- Thanks for engaging with us providing patches, suggestions, ideas, and reproducers
- The investment of the community is substantial and visible in the team composition and in our achievements
- We'll continue to do our best to enable your success continuing to deliver the best ROOT possible also during 2024

Backup

Core Team Members 2024

PC Philippe Canal (FNAL)

VV Vassil Vassilev (Princeton)

SL Serguei Linev (GSI)

MT Matevz Tadel (UCSD)

AM Alia Mrak Tadel (UCSD)

BB Bertrand Bellenot (CERN)

DP Danilo Piparo (CERN)

DS Devajith Valaparambil Sreeramaswamy (CERN, SYCLOPS)

FdG Florine de Geus (CERN and University of Twente)

JB Jakob Blomer (CERN)

JH Jonas Hanfeld (CERN and Goethe University of Frankfurt)

JR Jonas Rembser (CERN)

LM Lorenzo Moneta (CERN)

MC Marta Czurylo (CERN)

MD Monica Dessole (CERN, SYCLOPS)

OC Olivier Couet (CERN)

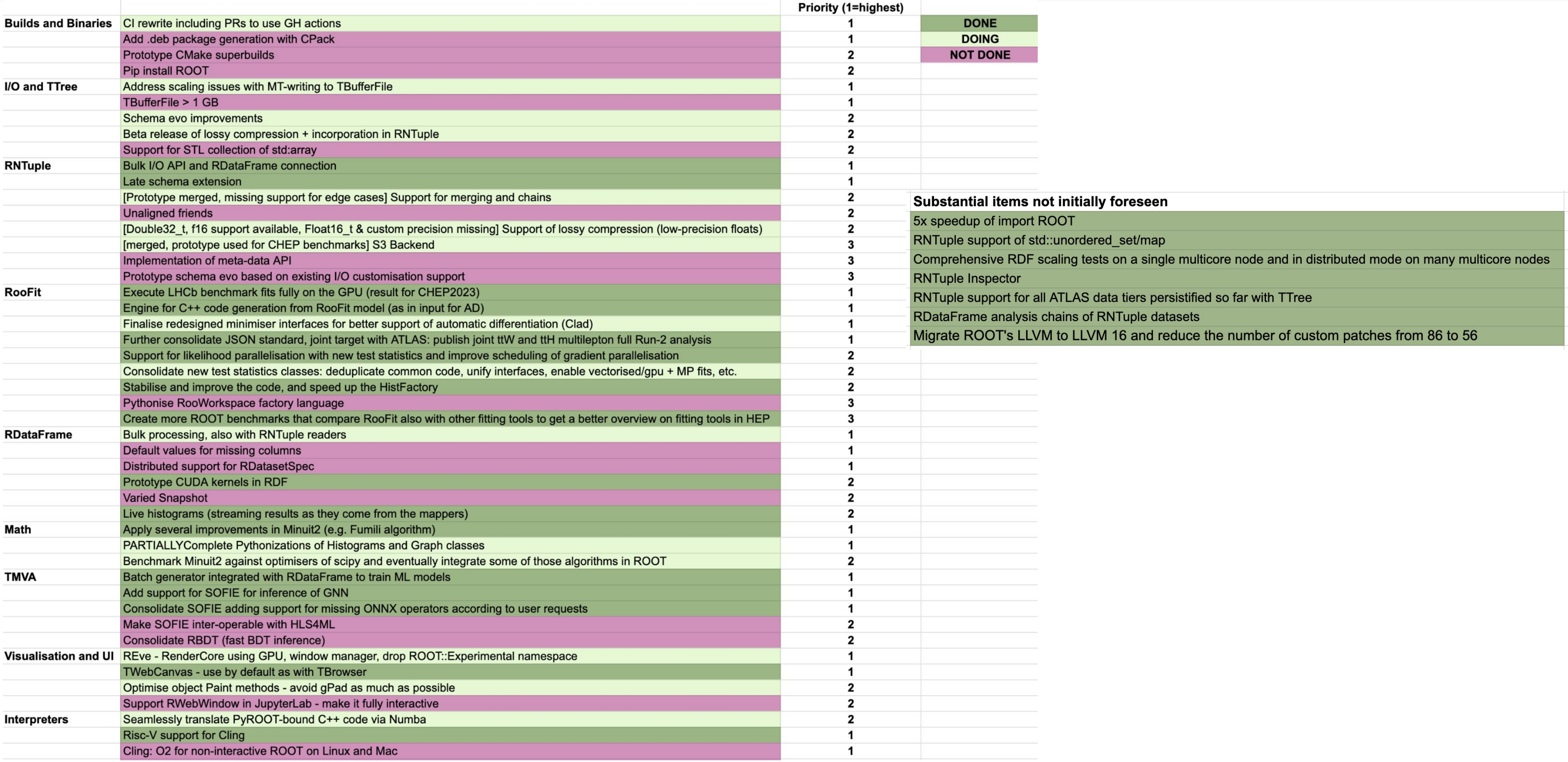
VP Vincenzo Padulano (CERN)

This list includes PhD students but not other kind of students, e.g. trainees or other kind of short term internships. This list does not reflect the many precious contributions to the ROOT project by our user community.

Stretch Goals

- I/O: Support for STL Collection of std::array
- RDF: hybrid scheduling: DistRDF + IMT
- RDF: distributed support for RDatasetSpec
- RDF: add default values for missing columns
- RDF: investigate the GroupBy functionality
- Builds: Provide CMake Superbuilds
- RNTuple: Different column-representation per cluster (in case of schema evolution)
- RNTuple: Design and implementation of virtual (derived) columns
- PyROOT: Initial assessment of the consequences of lifting of the Python GIL and compile an initial set of actions to be taken

2023 PoW Status



		Priority (1=highest)	
Builds and Binaries	▨ CI rewrite including PRs to use GH actions	1	🟩 DONE
	🟥 Add .deb package generation with CPack	1	▨ DOING
	🟥 Prototype CMake superbuilds	2	🟥 NOT DONE
	🟥 Pip install ROOT	2	
I/O and TTree	▨ Address scaling issues with MT-writing to TBufferFile	1	
	🟥 TBufferFile > 1 GB	1	
	▨ Schema evo improvements	2	
	▨ Beta release of lossy compression + incorporation in RNTuple	2	
	🟥 Support for STL collection of std:array	2	
RNTuple	🟩 Bulk I/O API and RDataFrame connection	1	
	🟩 Late schema extension	1	
	▨ [Prototype merged, missing support for edge cases] Support for merging and chains	2	
	🟥 Unaligned friends	2	
	▨ [Double32_t, f16 support available, Float16_t & custom precision missing] Support of lossy compression (low-precision floats)	2	
	▨ [merged, prototype used for CHEP benchmarks] S3 Backend	3	
	🟥 Implementation of meta-data API	3	
	🟥 Prototype schema evo based on existing I/O customisation support	3	
RooFit	🟩 Execute LHCb benchmark fits fully on the GPU (result for CHEP2023)	1	
	🟩 Engine for C++ code generation from RooFit model (as in input for AD)	1	
	▨ Finalise redesigned minimiser interfaces for better support of automatic differentiation (Clad)	1	
	🟩 Further consolidate JSON standard, joint target with ATLAS: publish joint ttW and ttH multilepton full Run-2 analysis	1	
	🟩 Support for likelihood parallelisation with new test statistics and improve scheduling of gradient parallelisation	2	
	▨ Consolidate new test statistics classes: deduplicate common code, unify interfaces, enable vectorised/gpu + MP fits, etc.	2	
	🟩 Stabilise and improve the code, and speed up the HistFactory	2	
	🟥 Pythonise RooWorkspace factory language	3	
	▨ Create more ROOT benchmarks that compare RooFit also with other fitting tools to get a better overview on fitting tools in HEP	3	
RDataFrame	▨ Bulk processing, also with RNTuple readers	1	
	🟥 Default values for missing columns	1	
	🟥 Distributed support for RDatasetSpec	1	
	🟩 Prototype CUDA kernels in RDF	2	
	🟥 Varied Snapshot	2	
	🟩 Live histograms (streaming results as they come from the mappers)	2	
Math	🟩 Apply several improvements in Minuit2 (e.g. Fumili algorithm)	1	
	▨ PARTIALLYComplete Pythonizations of Histograms and Graph classes	1	
	▨ Benchmark Minuit2 against optimisers of scipy and eventually integrate some of those algorithms in ROOT	2	
TMVA	🟩 Batch generator integrated with RDataFrame to train ML models	1	
	🟩 Add support for SOFIE for inference of GNN	1	
	🟩 Consolidate SOFIE adding support for missing ONNX operators according to user requests	1	
	🟥 Make SOFIE inter-operable with HLS4ML	2	
	🟥 Consolidate RBDT (fast BDT inference)	2	
Visualisation and UI	▨ REve - RenderCore using GPU, window manager, drop ROOT::Experimental namespace	1	
	🟩 TWebCanvas - use by default as with TBrowser	1	
	▨ Optimise object Paint methods - avoid gPad as much as possible	2	
	🟥 Support RWebWindow in JupyterLab - make it fully interactive	2	
Interpreters	▨ Seamlessly translate PyROOT-bound C++ code via Numba	2	
	🟩 Risc-V support for Cling	1	
	🟥 Cling: O2 for non-interactive ROOT on Linux and Mac	1	

Substantial items not initially foreseen
🟩 5x speedup of import ROOT
🟩 RNTuple support of std::unordered_set/map
🟩 Comprehensive RDF scaling tests on a single multicore node and in distributed mode on many multicore nodes
🟩 RNTuple Inspector
🟩 RNTuple support for all ATLAS data tiers persistified so far with TTree
🟩 RDataFrame analysis chains of RNTuple datasets
🟩 Migrate ROOT's LLVM to LLVM 16 and reduce the number of custom patches from 86 to 56