

ROOT Workshop 2022 Summary

Axel Naumann axel@cern.ch for the ROOT team
2022-05-23



Presentations, Slides, soon: Recordings

<https://indico.fnal.gov/event/23628/timetable/#all.detailed>

ROOT Users Workshops

- 12th event in a long tradition; previous: Sarajevo, 3.5 years ago
- Traditionally: inviting place, special setup, and budget to invite physics students over
- Plan for 2021: Fermilab: 1st institutional ROOT user, since almost 25 years
 - Fermilab and Chicago, with tours and wonderful dinner
- But... reality.

Layout of 2022 ROOT Workshop

- 250 registered participants, out of which 150 joined
- Virtual, with in-person meeting rooms at Fermilab (host) and CERN
- 3 half-days filled to the brim with high-quality contributions
 - From users, experiments, and ROOT devs
 - Impressive amount of preparation went into these talks!
 - 38 talks, super dense, virtually everything < 20 minutes

Presentations

- ROOT news
- Novice users reporting how they learned ROOT and what the problems were
- Expert users reporting on their use cases and wishes
- Almost a whole day of experiments presenting their (future) analysis models - DUNE, Compass, Mu2e, and of course LHC!
- Packagers presented challenges and solutions, e.g. conda, pip, snap
- And fun things: cling for music, ROOT for finance

<Interlude> ROOT plans (as presented at the workshop)

ROOT's Role

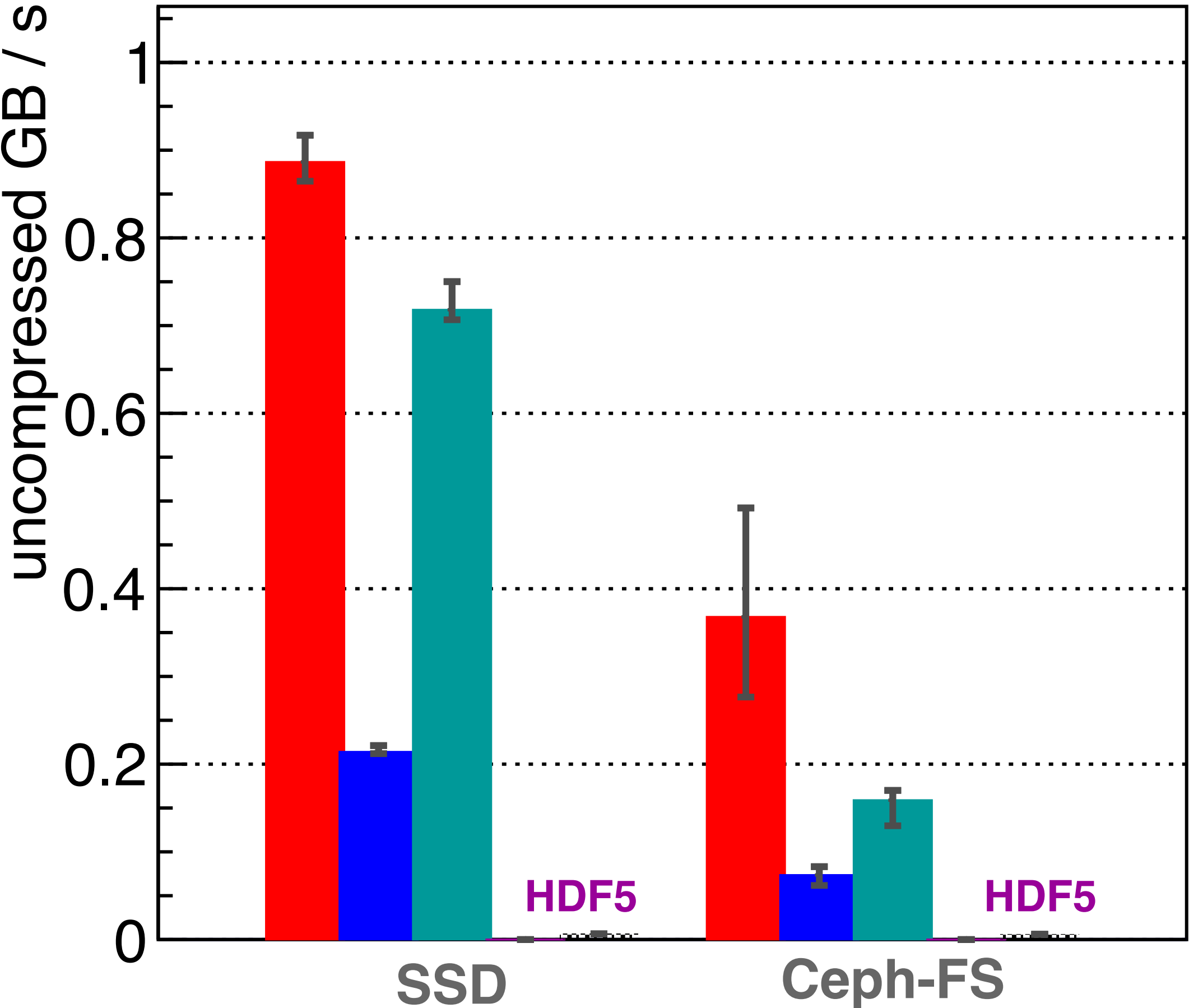
- ROOT is relied on by virtually all HEP experiments
 - We know since day 1, ROOT could be way more relevant outside HEP
- More than 1 exabyte of data is entrusted with ROOT
- ROOT serves as a HEP standard library, as a distribution mechanism, as the hub of an ecosystem
- Experts know ROOT and know how to use it - it's the new physicists where ROOT needs to convince

ROOT's Core Duties

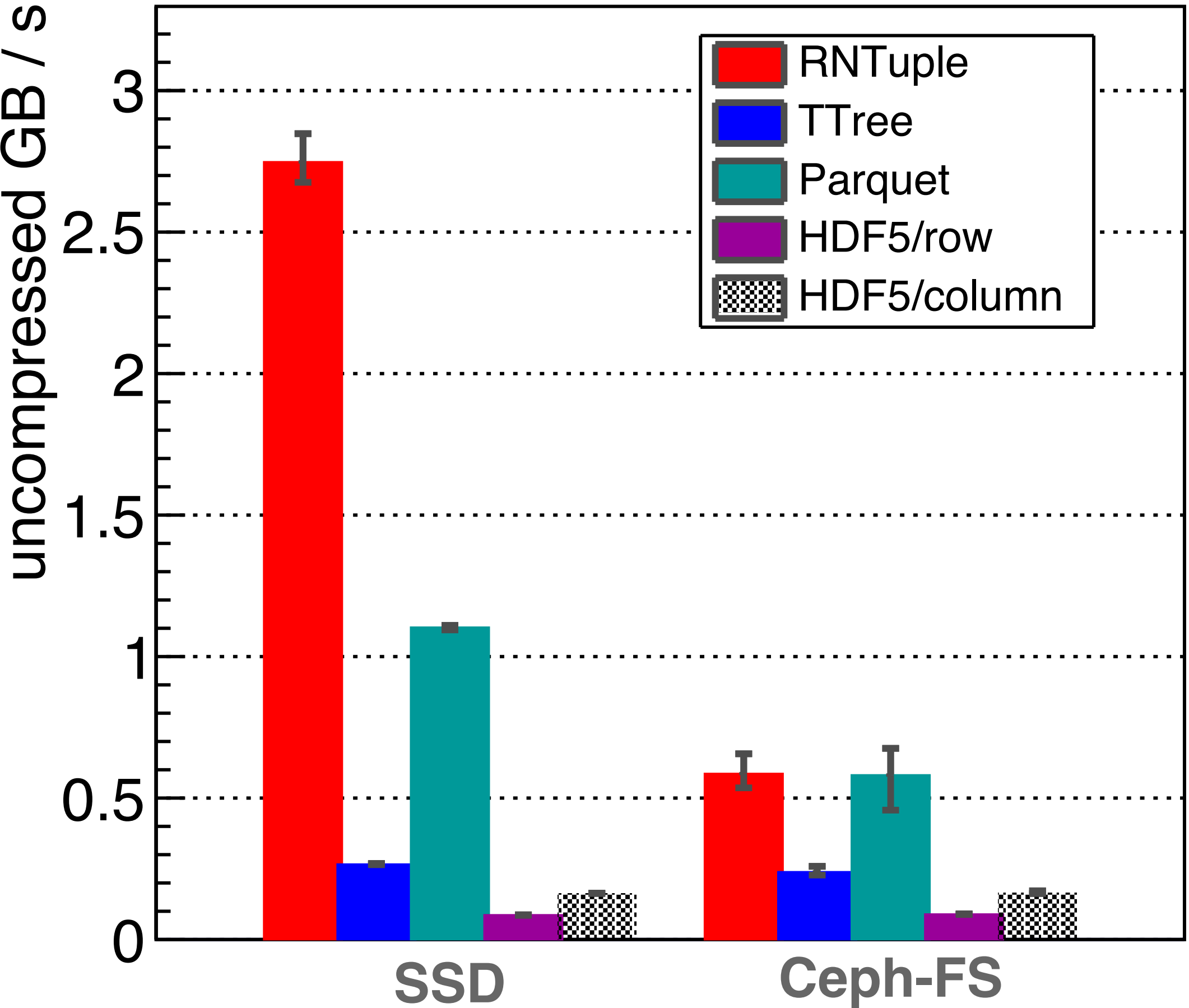
- Statistics and modeling: e.g. fitting, likelihood parametrization
- I/O: e.g. reading (+ writing) data, efficiently
- Math library: e.g. PRNG, Lorentz vector, differentiation
- Analysis interfaces: e.g. RDataFrame, histograms
- Efficient interplay with machine learning libraries
- Graphics: e.g. plots on your screen, publication-grade scientific visualization, event display

What we have

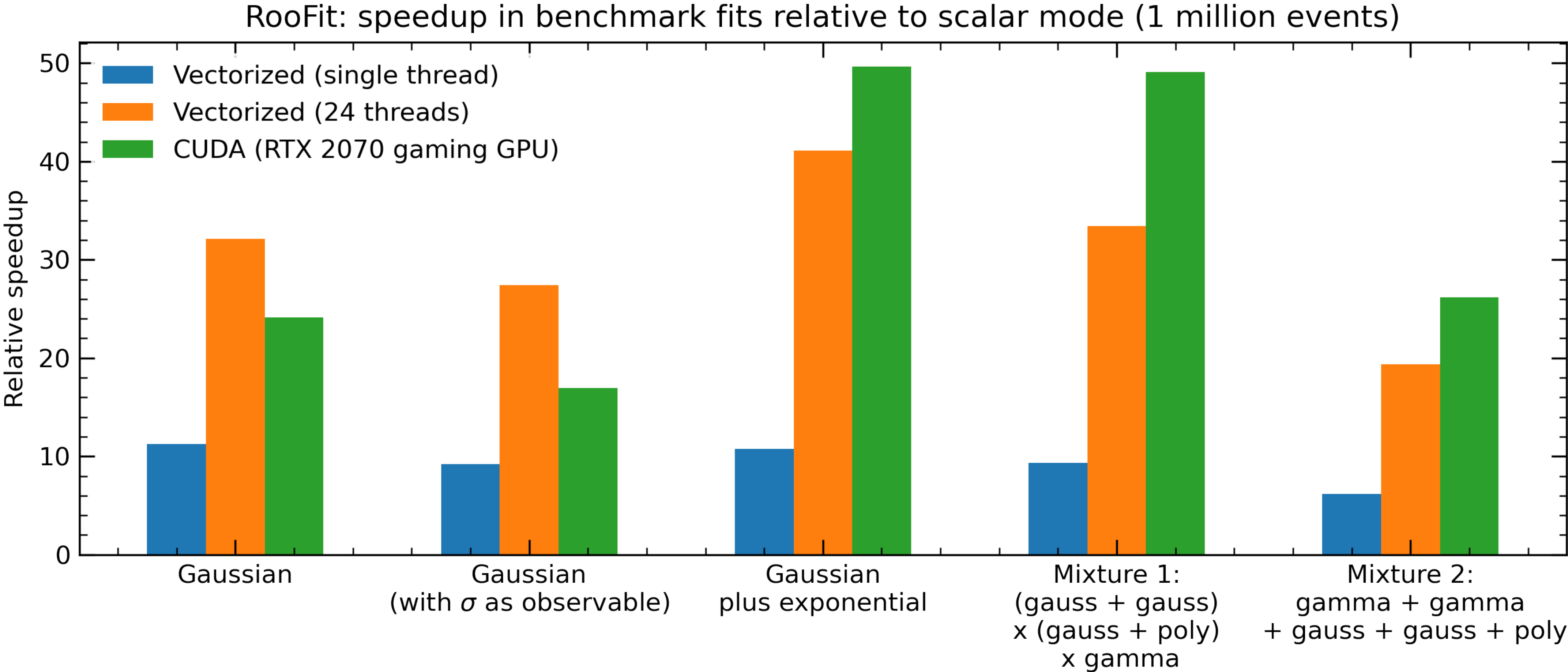
CMS Higgs4Leptons (10/84 branches)



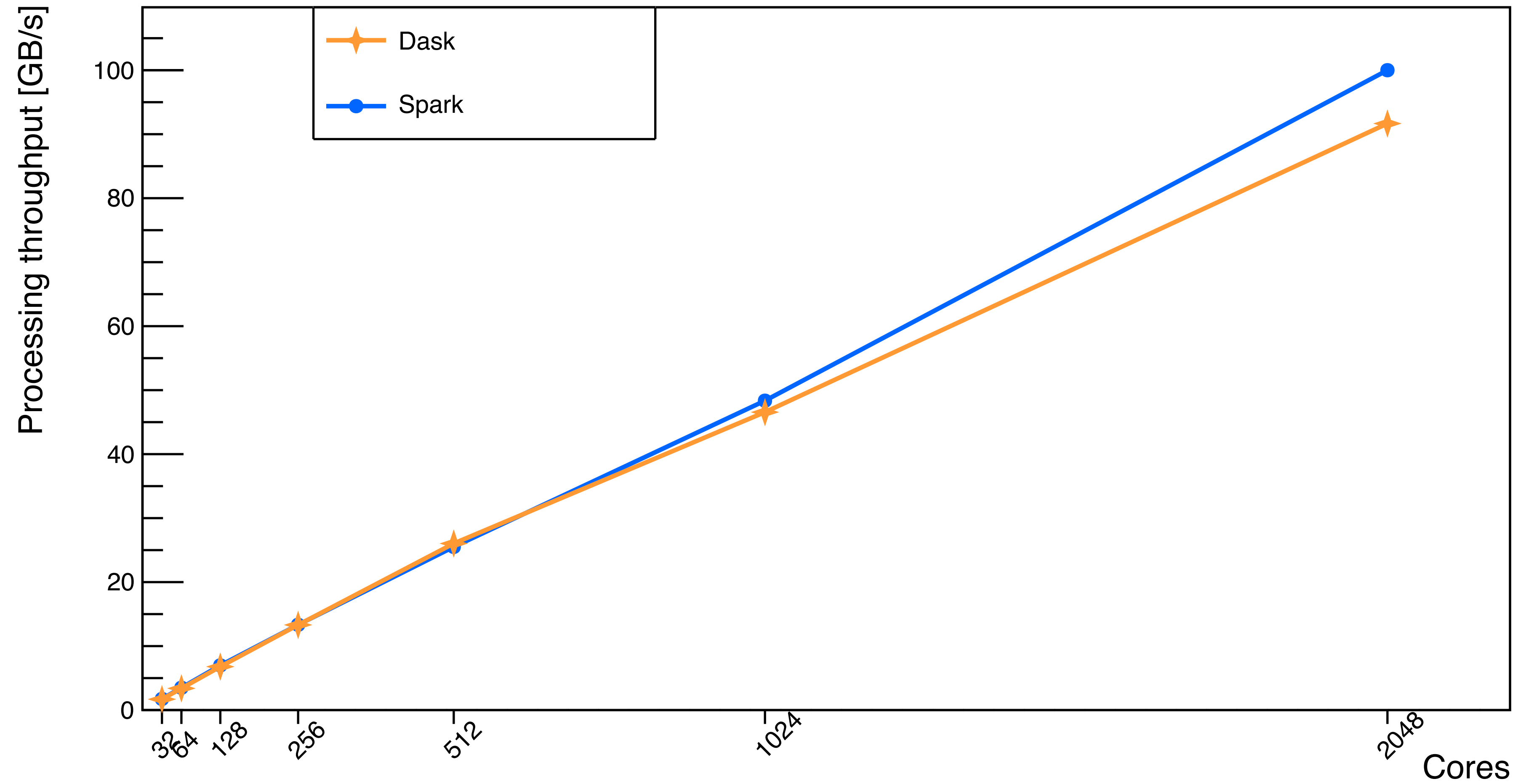
LHCb B2HHH (10/26 branches)



What we have



What we have



ROOT's Plans

- Unbeatable throughput for piping data into machine learning frameworks, training and inference (e.g. SOFIE)
- Versatile, understandable, robust, and highly efficient analysis interface
RDataFrame
- Providing the I/O format for the next 25 years
- A simple high-level graphics system: plot trigger efficiency! Draw a 2D histogram with categories!
- Well-designed interplay between I/O, ML, graphics, histograms, RDataFrame

Don't we have that "Today"?

- Still a long way:
 - Integration of different parts
 - Reaping their benefits



CC BY-SA 3.0, <https://commons.wikimedia.org/w/index.php?curid=1762839>

- We are building the pillars, the bridges are next. R&D + engineering!
- ROOT of 2025: easy to use; with a consistent, coherent design, as always

import ROOT

- ROOT as a seamless Python module, for everything we do
- Explicit goal: excellent integration and interoperability
- Reap the benefits of a fast C++ core, without exposing that C++ core

What Else? Major R&D Topics

- Next-generation histograms: simple design; categorical, circular, counting axes; multiple uncertainties; multiple weights per bin; well integrated with RDataFrame, I/O, graphics
- Auto-diff and GPUize the world: faster computations / minimization
- Simpler install: download mini-ROOT, grab the rest as needed.
(c) Rene Brun ("BOOT"), 15 years ago...
- C++ reflection instead of dictionaries, simpler TClass
- (Distributed) RDataFrame: internal optimization of analysis, incl. GPU offloading

Future Challenges

- We see that C++ is the right core: stable, flexible path towards high **efficiency**. Yet, fewer students know C++. Not a new fear nor specific to ROOT, and we manage to counter it by attracting brilliant computer scientists
- Continuing separation of expert-level ROOT implementation, from user-level ROOT interfaces: harder to contribute for physicists, harder to "make it yours". ROOT becomes like Linux (install and use), instead of "a tool by us, for us"
- Significant investment in "a world without ROOT" (as a goal by itself), provides fantastic occasions for benchmarking against alternatives, and show the community why to trust ROOT

</Interlude>

Was the Workshop Useful?

- Many presentations, all too short!
- High quality feedback:
 - not just praise but also criticism, food for thought, ideas
- Asking excellent speakers to "just say what you think" brought useful / surprising feedback to the table
- Sharing ROOT news was hopefully useful, too: about 140 participants on Monday

Train the Trainer

- Training event after 3 workshop half-days
 - 2nd ROOT train the trainer event, to enable sharing of material and experience between ROOT team and non ROOT-team trainers
- 25 participants
- ROOT team: "that's our material"; non-ROOT trainers: "that's our problems"
- Excellent presentations by trainers, e.g. from experiments and Bill Seligman, training ROOT since 20 years!

Train the Trainer: conclusion

- Introduction to programming: ROOT should simply provide links to existing trainings (C++ and Python). Do other projects have that demand, too?
- Useful existing material, though no common approach: RDataFrame or TTree? Old interfaces still play a major role in training; slow movement to new ones.
- Jupyter is the training standard, but generally not seen as sufficient for analyses and thus prompt etc often part of training. GUI highly relevant for training and students, e.g. fit panel.
- ROOT from Python or C++? Both seem to be wanted.

Specific feedback / observations

- RDataFrame is everywhere; analysis facilities take it into account
- Performance is now at a stage where focus shifts to ergonomics
 - `Vary()` was explicitly praised in that context
- RHist will solve a whole bucket of issues: interplay, C++ vs Python, performance, usability, multi-weight bin content
 - ROOT needs to deliver, sooner rather than later

Conclusion

- What we do has an impact: we can make users' life easier (or harder...)
- When we invest, users notice: good tutorials, good documentation
 - This is not a given; it means we invest where it counts
- We can do much more on histograms, ROOT's python face, usability
- Performance is good - usability is what really matters: Vary(), nD-histos, etc

Recordings

- Recordings currently being edited
- Will be accessible through <https://indico.fnal.gov/event/23628/timetable/#all.detailed>
- Several excellent contributions are certainly worth your time!



THANK YOU

for your caring about ROOT!

Martin Vorel, [CC BY-SA 4.0](#), via Wikimedia Commons

