ROOT 2022

Axel Naumann axel@cern.ch for the ROOT team

2022-05-09

3.5 years ago, at ROOT Workshop 2018, Sarajevo

- 45% new developers in the team
- Big new topics:
  - RDataFrame!
  - TTree-successor!
  - New web-graphics & web-GUI & event display!
  - Lots of news in RooFit + TMVA!

ROOT
Data Analysis Framework

FUN WITH DATA!

ROOT USERS WORKSHOP 2018
SARAJEVO, SEPTEMBER 10 - 13

WORKSHOP TOPICS

- ROOT ROADMAP
- TDATAFRAME, A MODERN TTREE::DRAW()
- ON THE WAY TO ROOT 7: GRAPHICS, HISTOGRAMS, TREES
- NEW PERSISTENCY FEATURES
- PARALLELISM IN AND WITH ROOT
- ROOT IN JAVASCRIPT
- MACHINE LEARNING IN AND WITH ROOT
- MATH TOOLS AND TECHNIQUES
- USER FEEDBACK

ORGANIZERS

WOLF BEHRENHOFF
PHILIPPE CANAL, FNAL
AXEL NAUMANN, CERN
EDMOND OFFERMANN
CINZIA PINZONI, CERN
DANILO PIPARO, CERN

CONTACT

CERN.CH/ROOT2018
ROOT-2018@CERN.CH

What does ROOT do in 3.5 years?

- In 2018, much of this was a seed: ideas, design discussions, R&D
- Estimated investment over 3.5 years:
  - RNTuple, TTree-successor: 1 developer
  - RDataFrame: 1.5 developers
  - Web graphics / GUI / event display: 1.5 developers
  - RooFit: 1.5 developers
- ROOT Workshop 2022: covers result of 3.5 years of creativity, work, and benchmarking

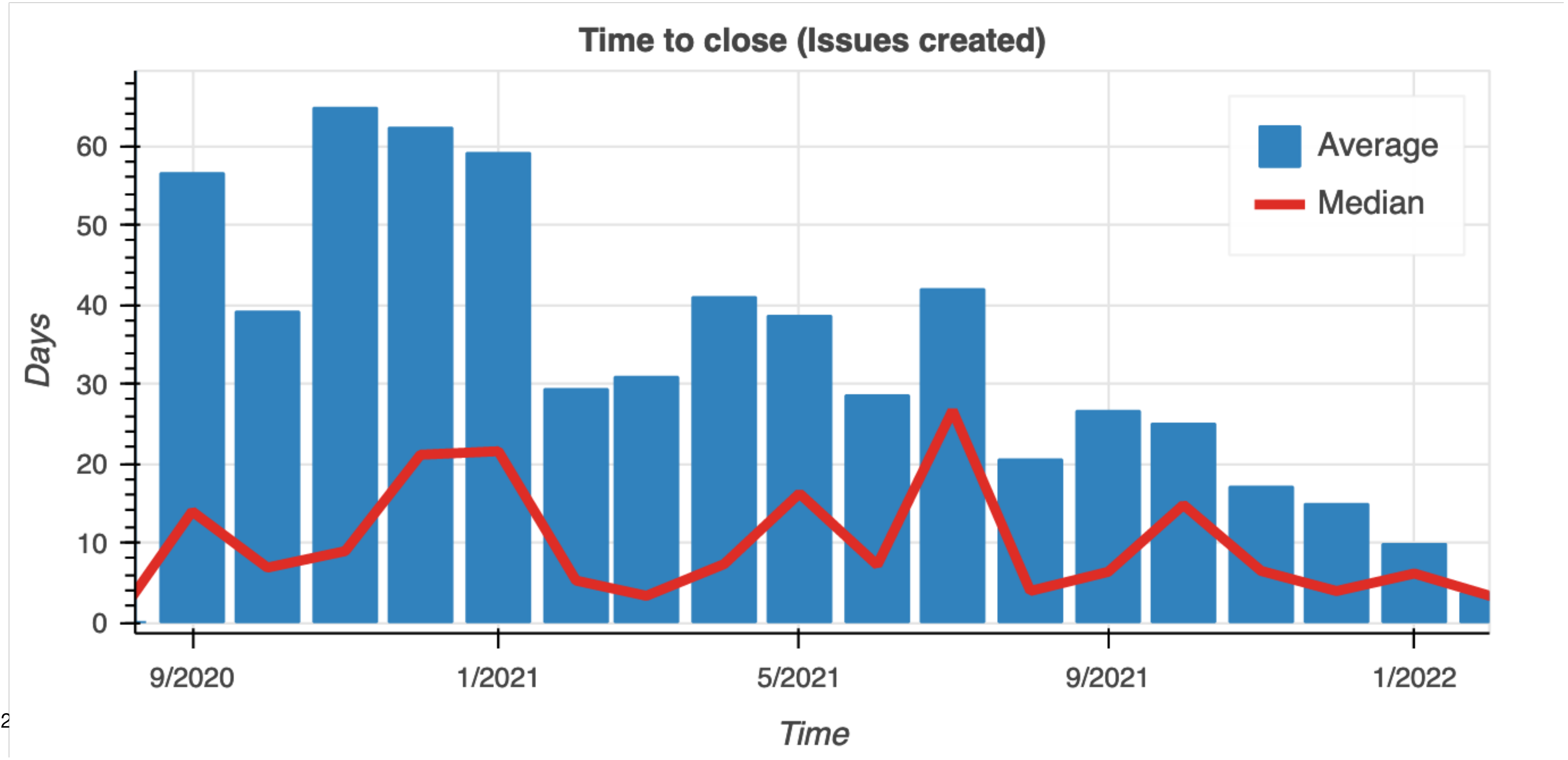
2021

Support!

- https://root-forum.cern.ch stats of 2021:
  - 15'000 posts, after 17k in 2020 and 14k in 2019
  - 1.4k new users, after 1.3k (2020) and 1.0k (2019)
  - 1st response on average after 11h, after 19h (2020), 30h (2019) [only topics with an answer are taken into account]
  - CERN's #1 forum
- We spend a considerable time here, web forum is virtually only channel

Bugs

- 50% of issues closed after about 10 days,
  all stats thanks to https://cauldron.io/project/5676



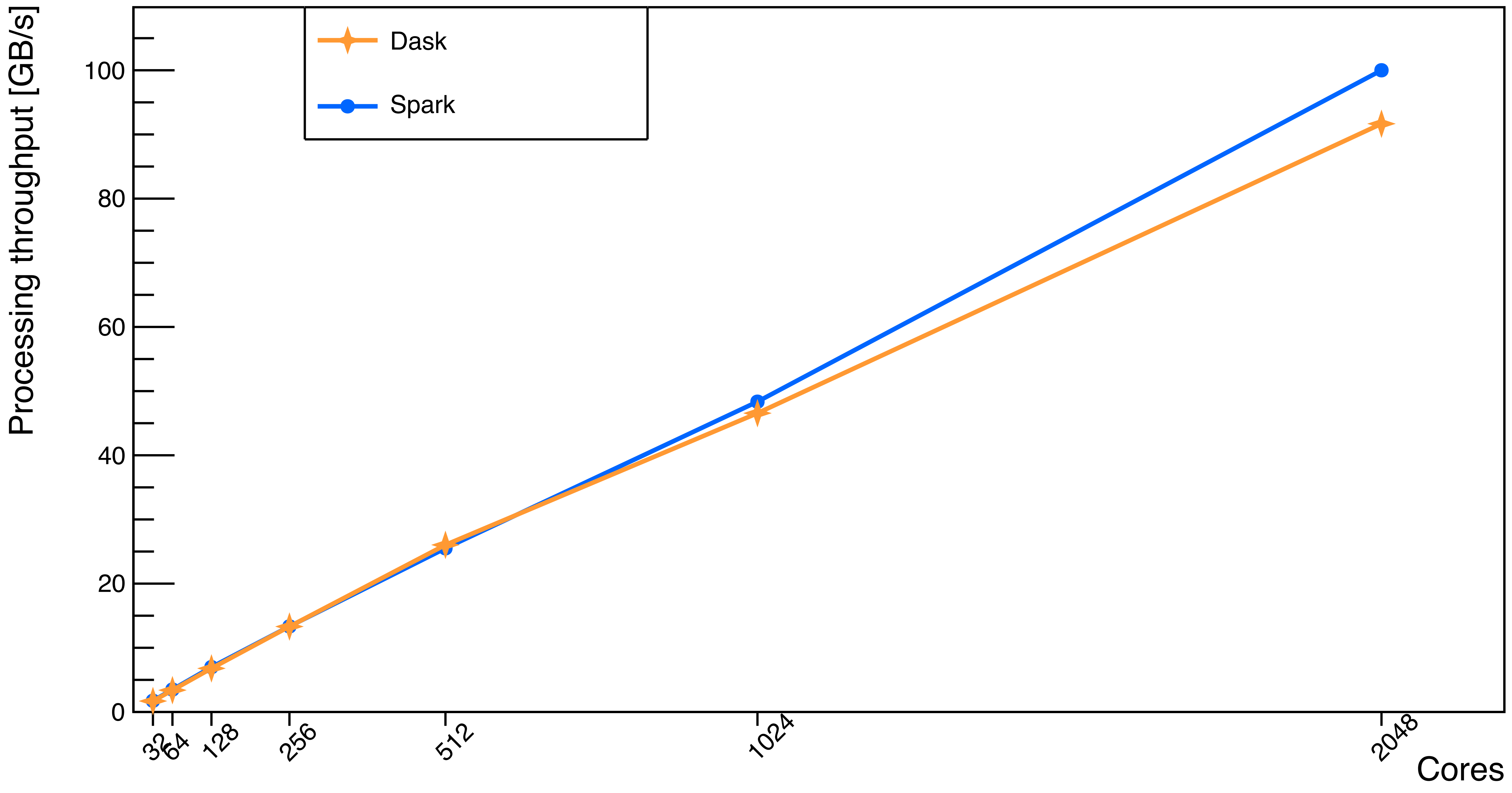
2021's Major Features

Distributed RDataFrame

- RDataFrame used by N*10% of analyses, sometimes embedded in analysis mini-frameworks: Bamboo, CROWN, Wmass,...
  - RDataFrame scales through multi-threading
- Distributed RDataFrame: scale across nodes (cluster), PROOF succession
  - Python-layer over RDataFrame: same interfaces, re-use of industry standard schedulers / cluster "adaptors": Dask (i.e. HTCondor etc), Spark, AWS Lambda

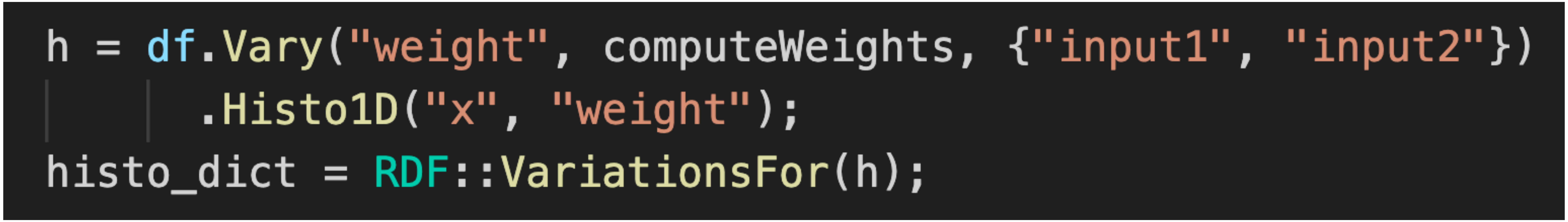
Distributed RDataFrame

- Prototype became minimal viable product in 2021
  - Feedback from physicists + first analysis groups are using it!
  - Lots of attention from the community: real demand
- Incorporating input from PROOF devs + experts, as well as cluster admins



RDataFrame::Vary()

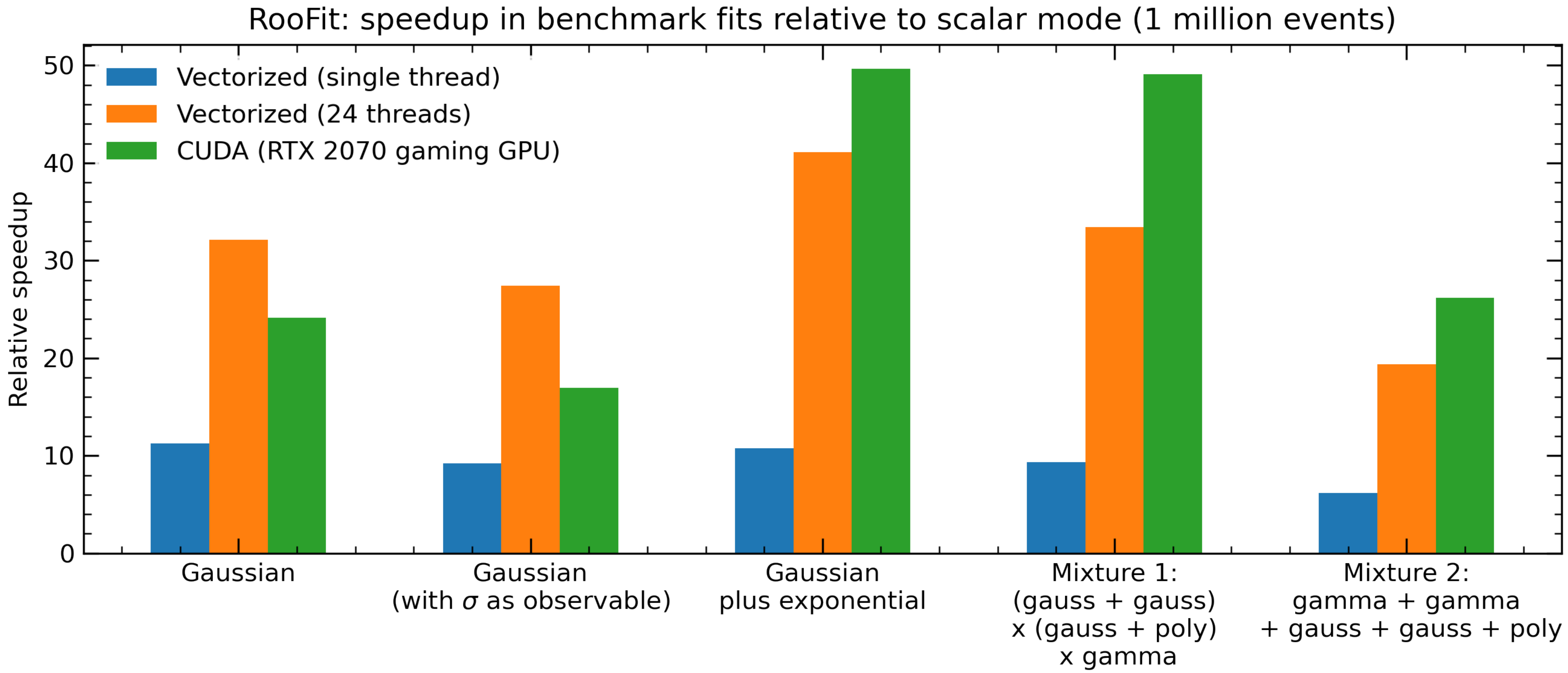
- Can be anywhere inside the whole analysis, anything: weight, input data, efficiency,...
- Creates a "parallel universe" of everything that depends on the varied value
- Evaluates everything in one single loop through data: a **game changer** behind an incredibly simple interface!



```
h = df.Vary("weight", computeWeights, {"input1", "input2"})
      .Histo1D("x", "weight");
histo_dict = RDF::VariationsFor(h);
```

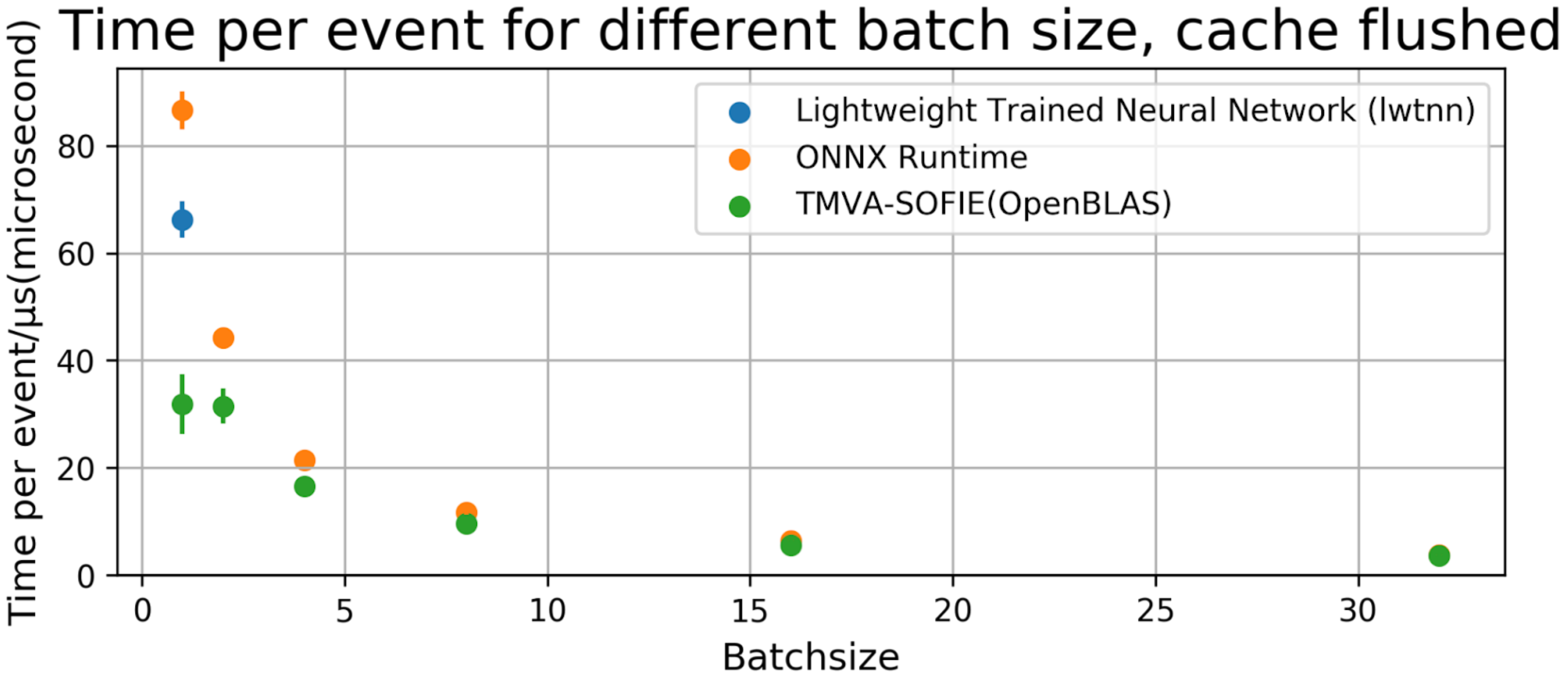
RooFit GPU + Pythonizations

- RooFit now has architecture-specific accelerator libraries for key functions
    - Optimal one loaded at runtime, given current architecture
    - Now also includes GPU version!
- Much improved Python interfaces!



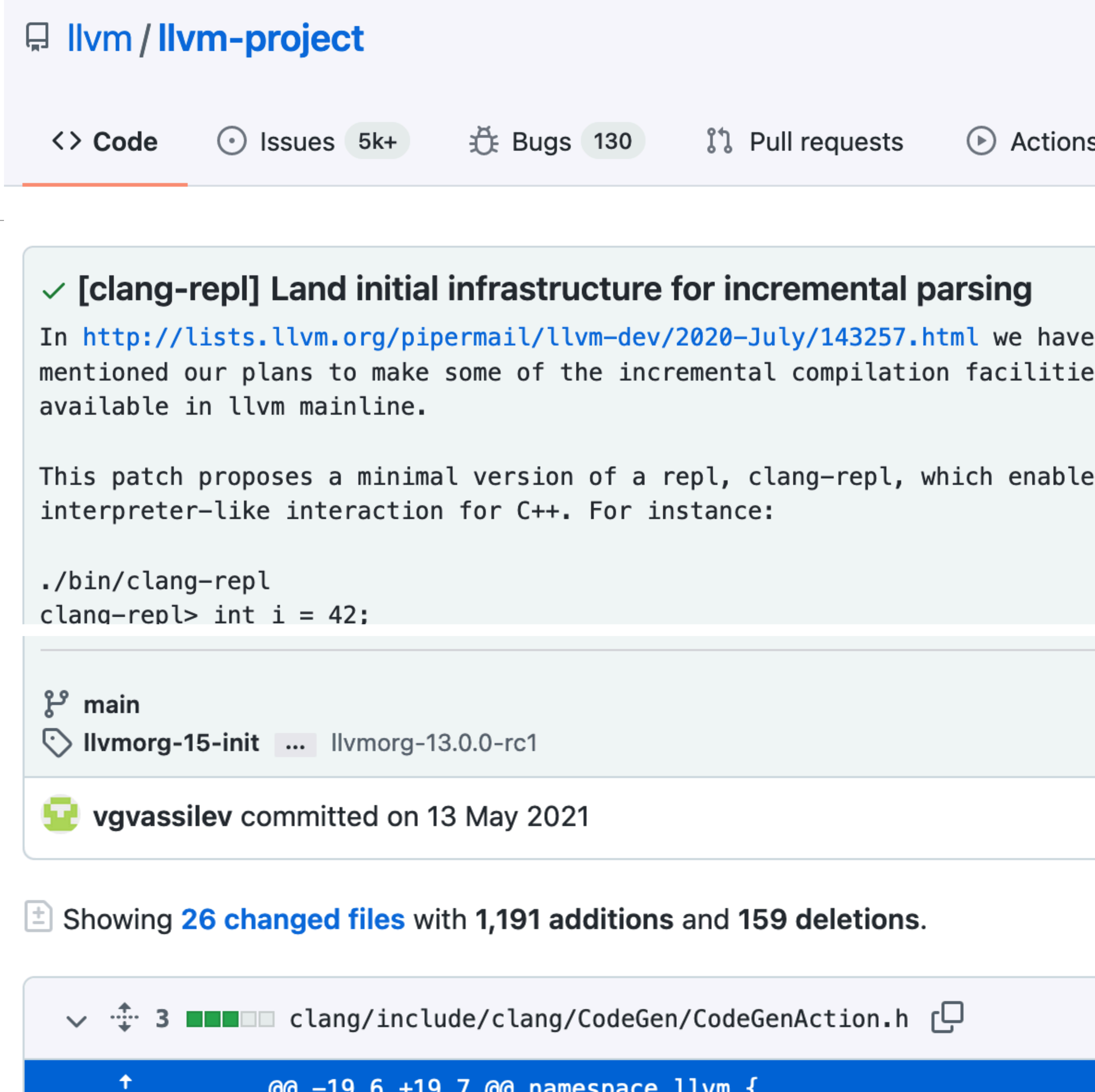
TMVA SOFIE

- ONNX is standard interchange / persistency format for trained models
- SOFIE can read those and generate C++
- Much more performant than ONNX runtime
- Incredibly lean
  (BLAS dependency)



Interpreter / Binding, Build

- clang-repl part of LLVM!



✓ [clang-repl] Land initial infrastructure for incremental parsing

```
In http://lists.llvm.org/pipermail/llvm-dev/2020-July/143257.html we have
mentioned our plans to make some of the incremental compilation facilitie
available in llvm mainline.

This patch proposes a minimal version of a repl, clang-repl, which enable
interpreter-like interaction for C++. For instance:

./bin/clang-repl
clang-repl> int i = 42;
```

main

llvmorg-15-init ... llvmorg-13.0.0-rc1

vgvassilev committed on 13 May 2021

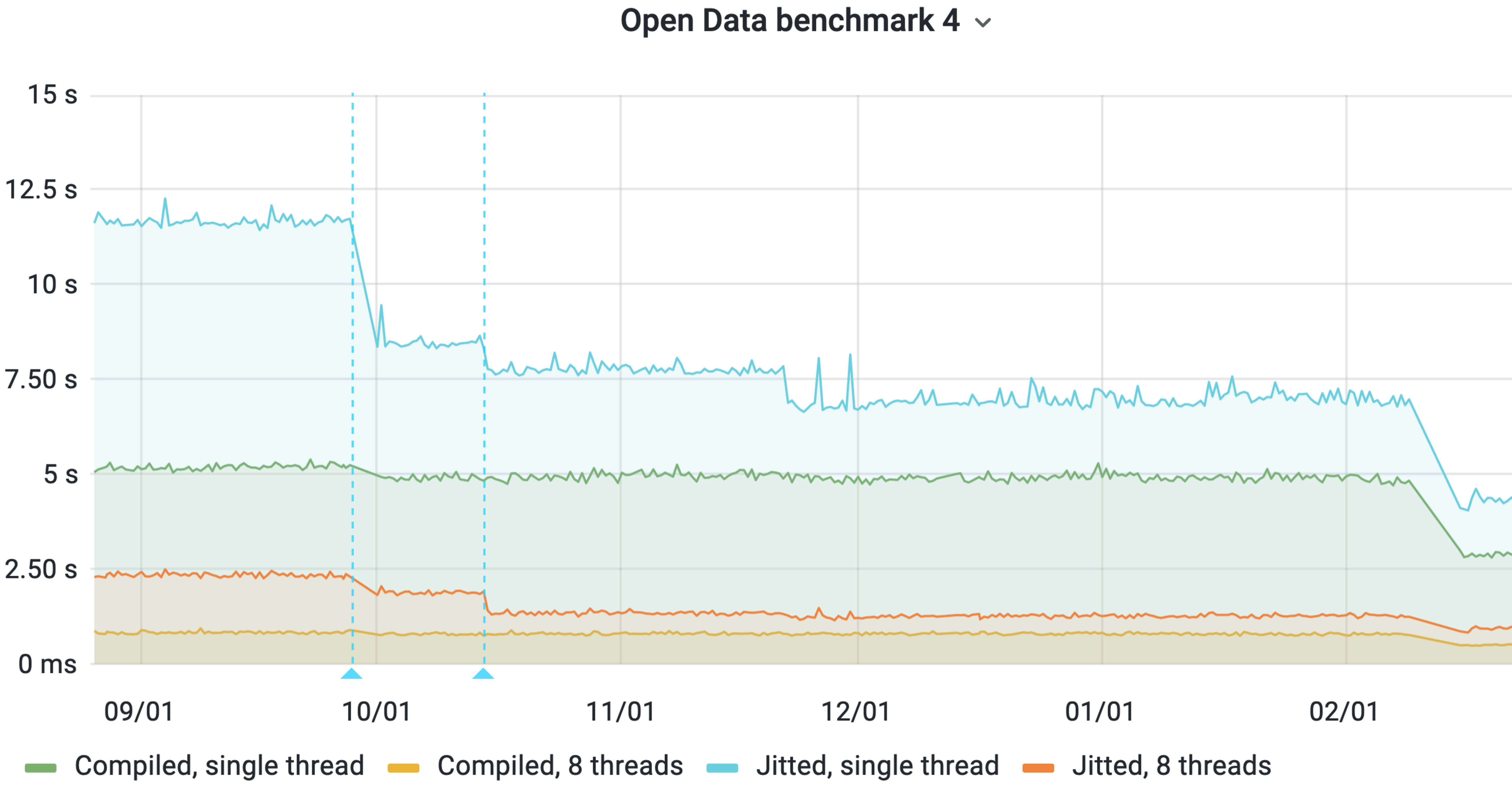
Showing 26 changed files with 1,191 additions and 159 deletions.

3 🟩🟩🟩□□ clang/include/clang/CodeGen/CodeGenAction.h

```
@@ -19,6 +19,7 @@ namespace llvm {
```

Interpreter / Binding, Build

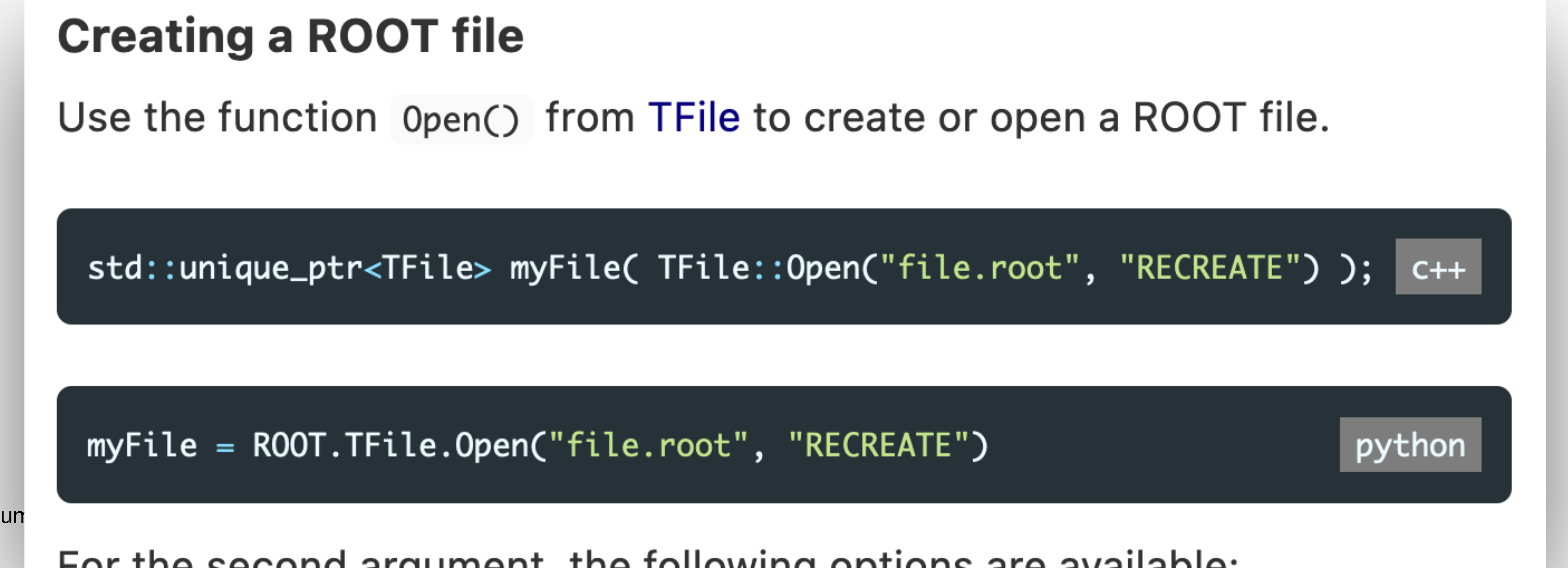
- Upgrade of cling to LLVM 9
    - ROOT now requires C++14
    - significant JIT optimization: "interpreted" code == compiled code



- ROOT has updated docker images, Conda nightlies

Documentation

- Team spent two weeks on documentation, manual
  - Complete re-write using modern ROOT, Python and C++
- Multiple blog posts, including contributed ones



Creating a ROOT file

Use the function `Open()` from TFile to create or open a ROOT file.

```c++
std::unique_ptr<TFile> myFile( TFile::Open("file.root", "RECREATE") );
```

```python
myFile = ROOT.TFile.Open("file.root", "RECREATE")
```

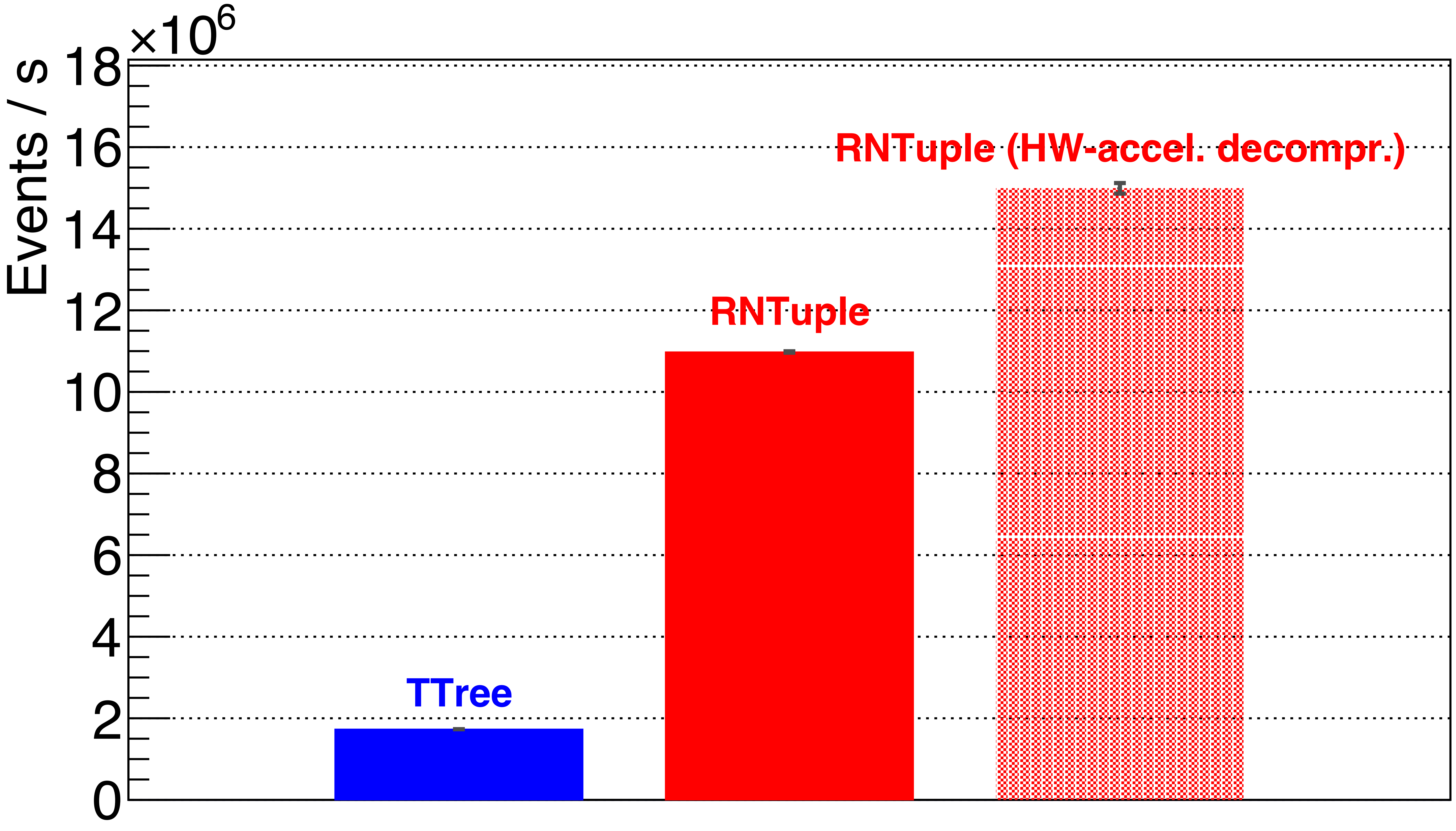
For the second argument, the following options are available:

RNTuple

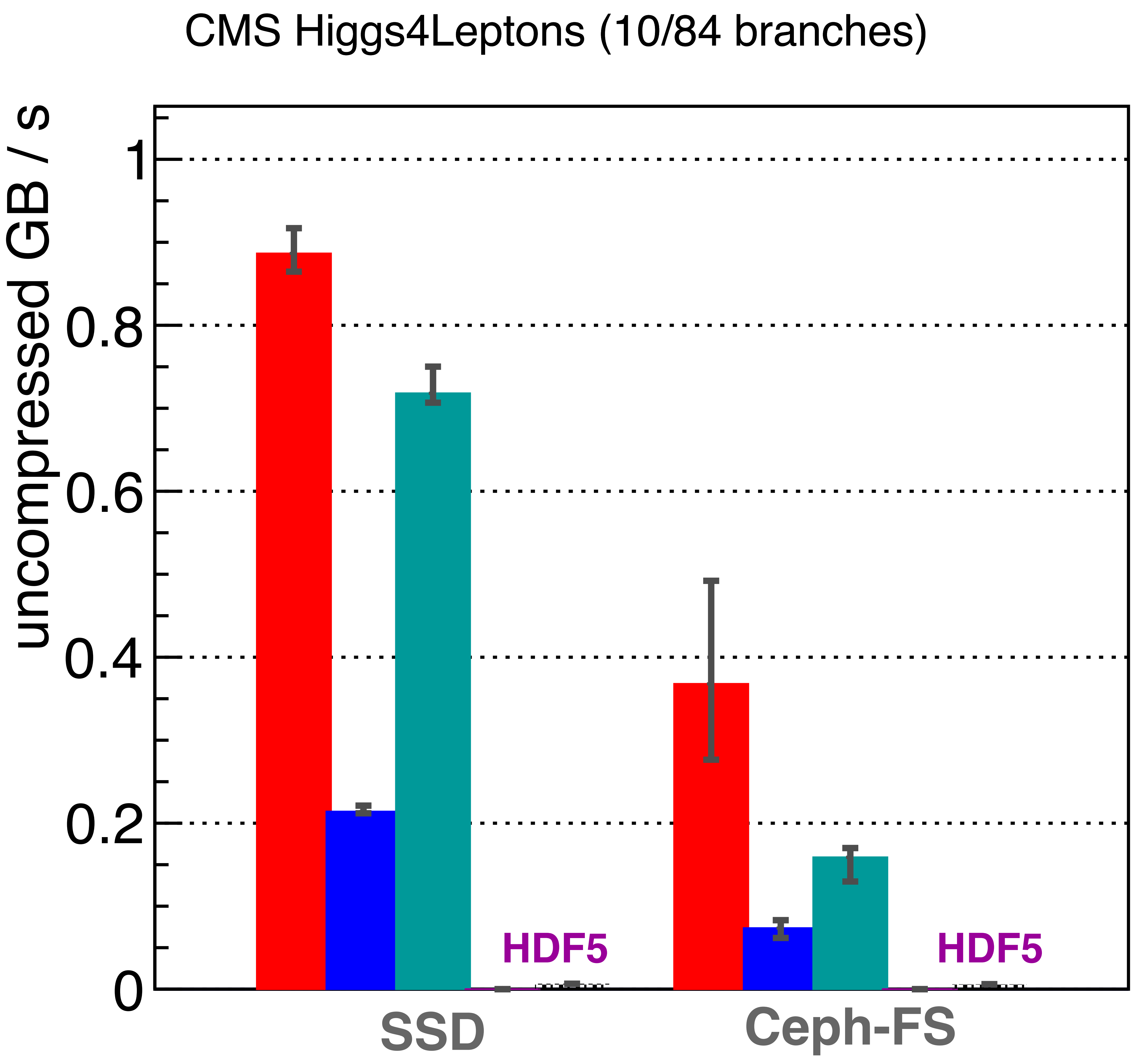
- A revolution: TTree successor, after 20 incredibly successful years
- Scheduled for production for HL-LHC
- Motivation: faster, smaller, future devices, sturdy, combining the best of 20 years of experience and state-of-the-art alternatives
- Result: **3-5x** faster, **-10..-20%** storage = 5..10MCHF/y

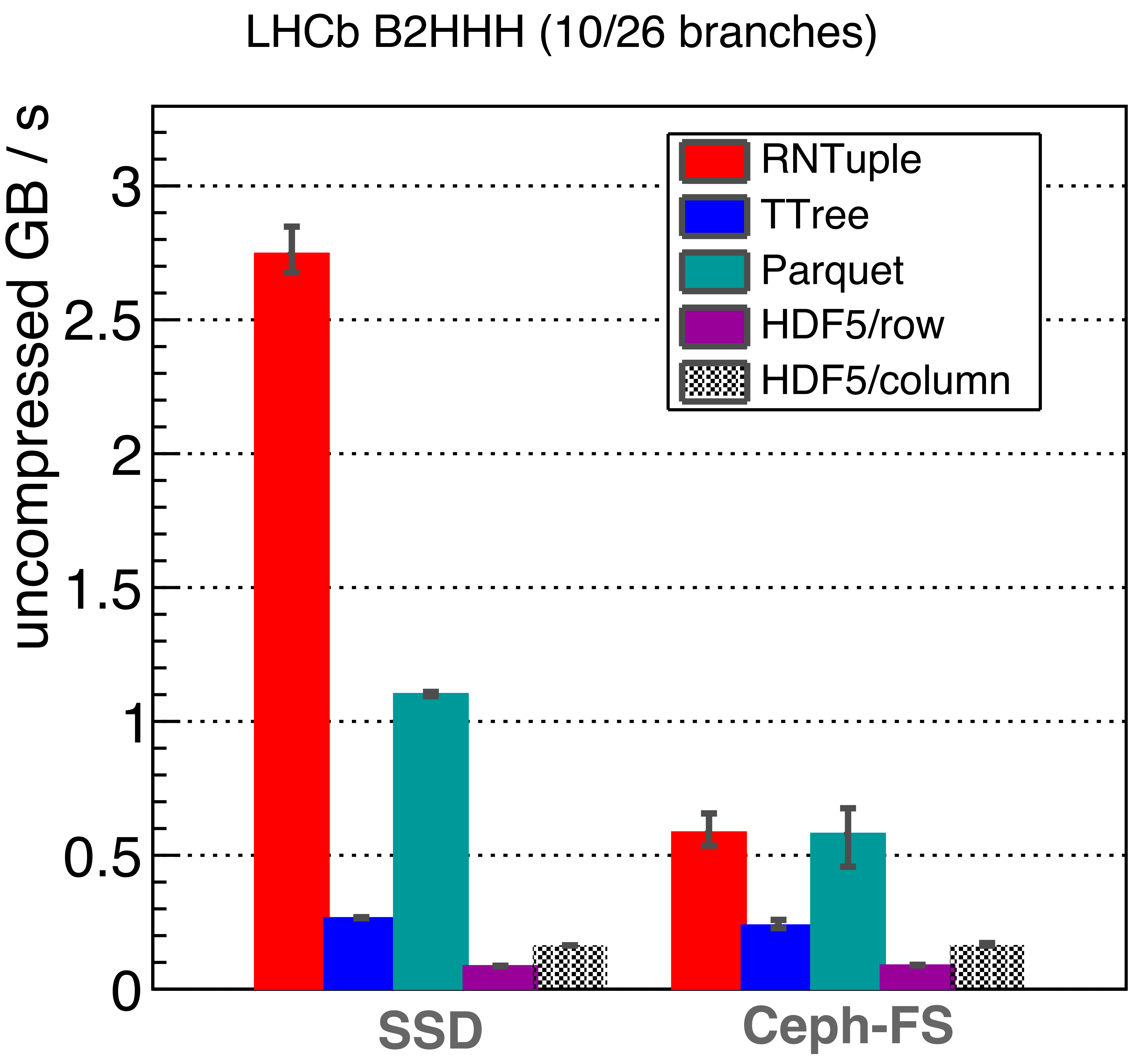
I/O Performance

- Read throughput SSD, Di-Muon Analysis on CMS nanoAOD



Analysis Performance

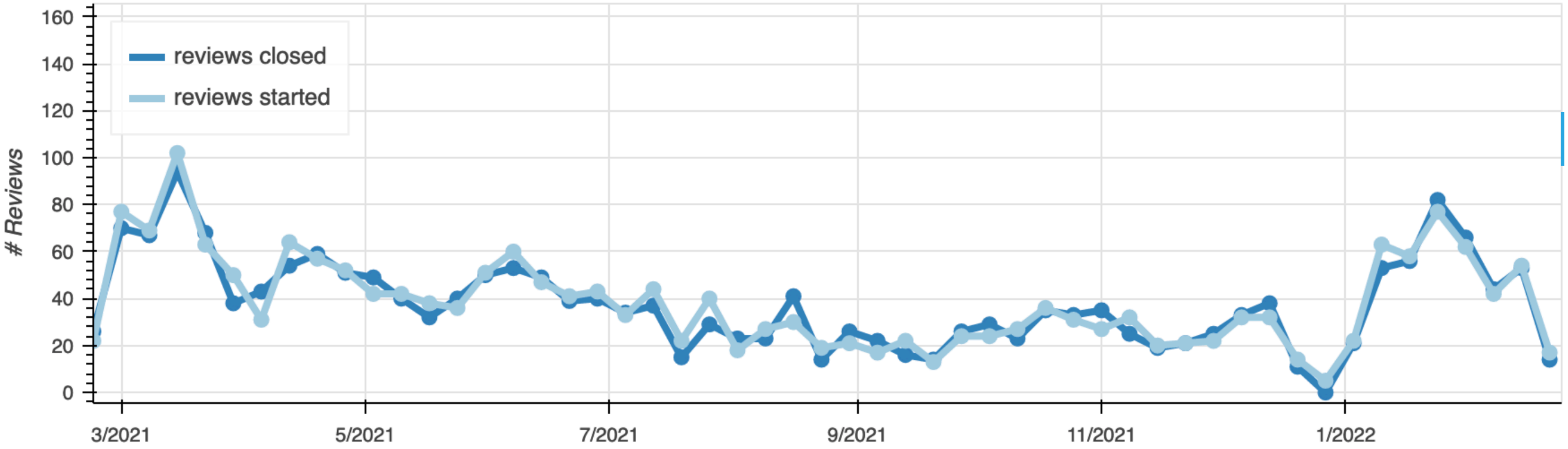




2021 Dev Statistics

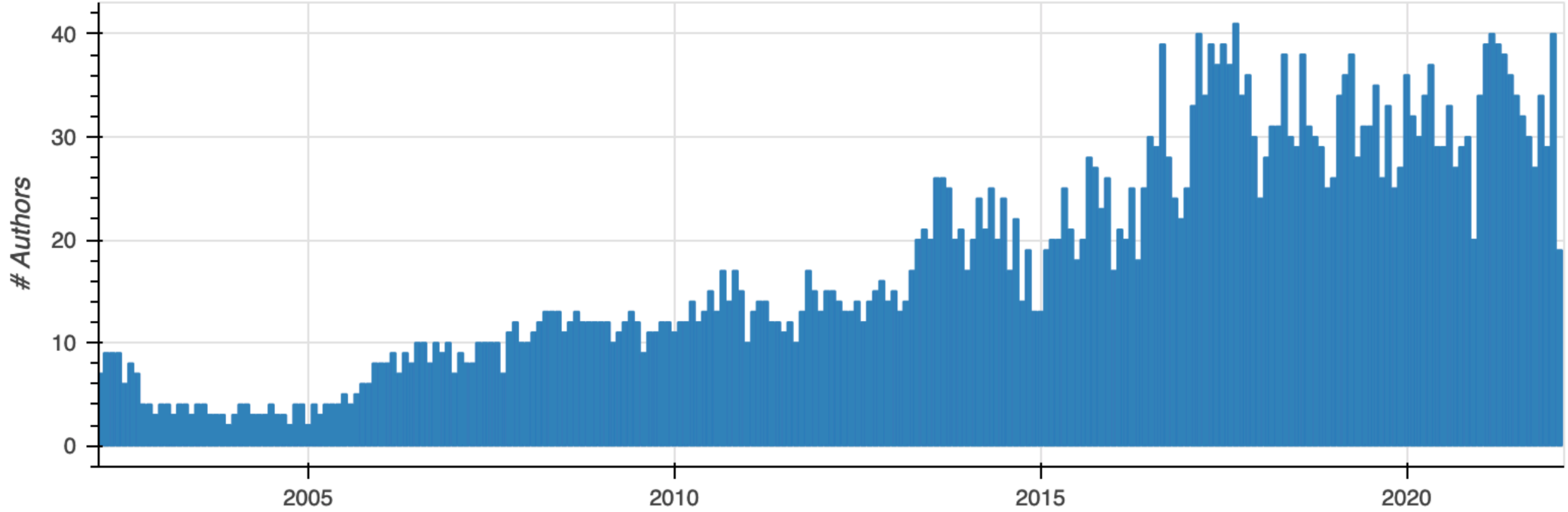
Code Change = Pull Requests

- About 2000 PRs over 2021, PRs per week:



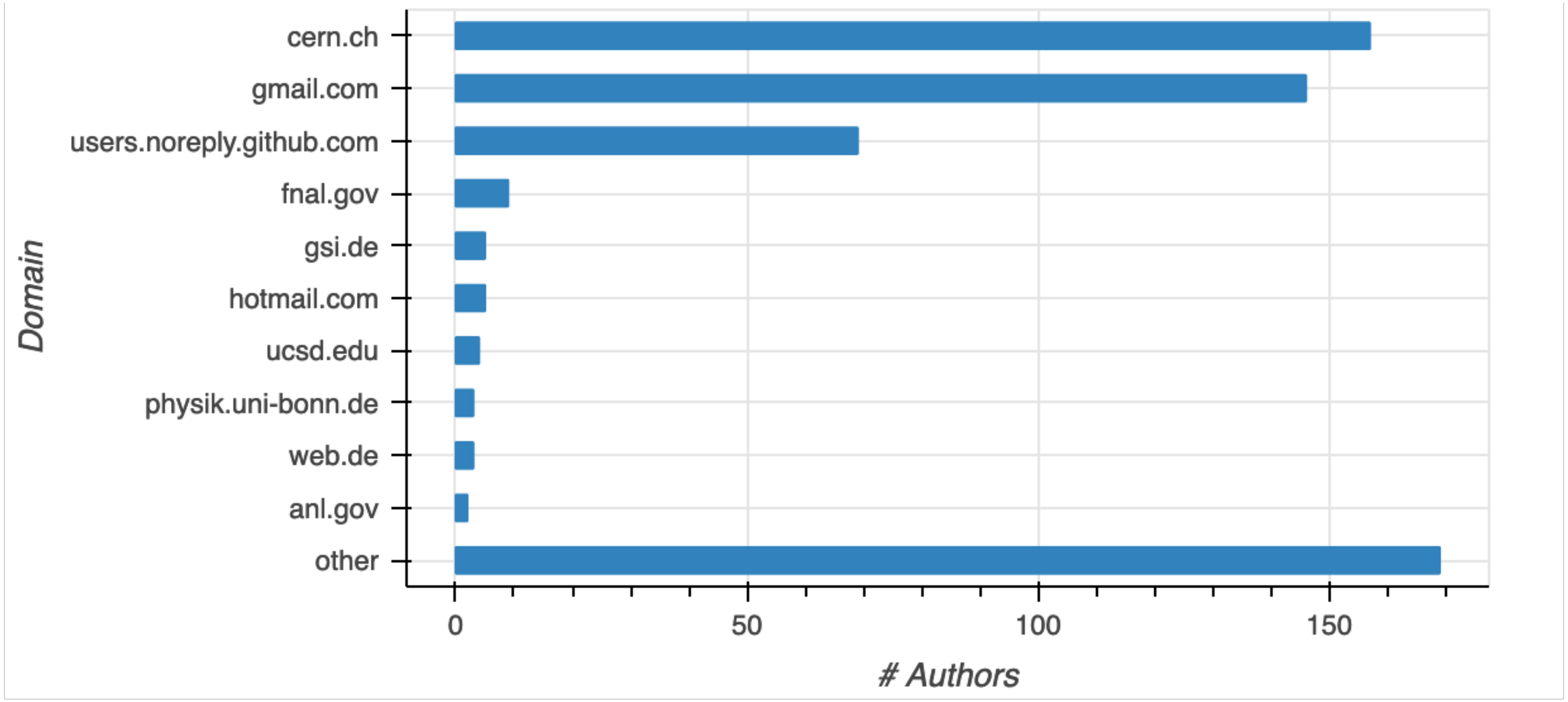
Contributors

- Consistently high number of contributors / month



Contributors

- High ratio of community contributors / month



Communication

Presentations, Working Groups

- Conferences: ACAT, vCHEP, EPS-HEP, LHCP, JLAB round table, PyHEP, Dask Distributed Summit, HIPS'21, CMMSE'21, CCGrid, ICTOpen, ICHEP

- Several presentations with experiments' physics groups

- Engagement with experiments, e.g. CMS analysis tools task force, ATLAS RooFit Hackathon

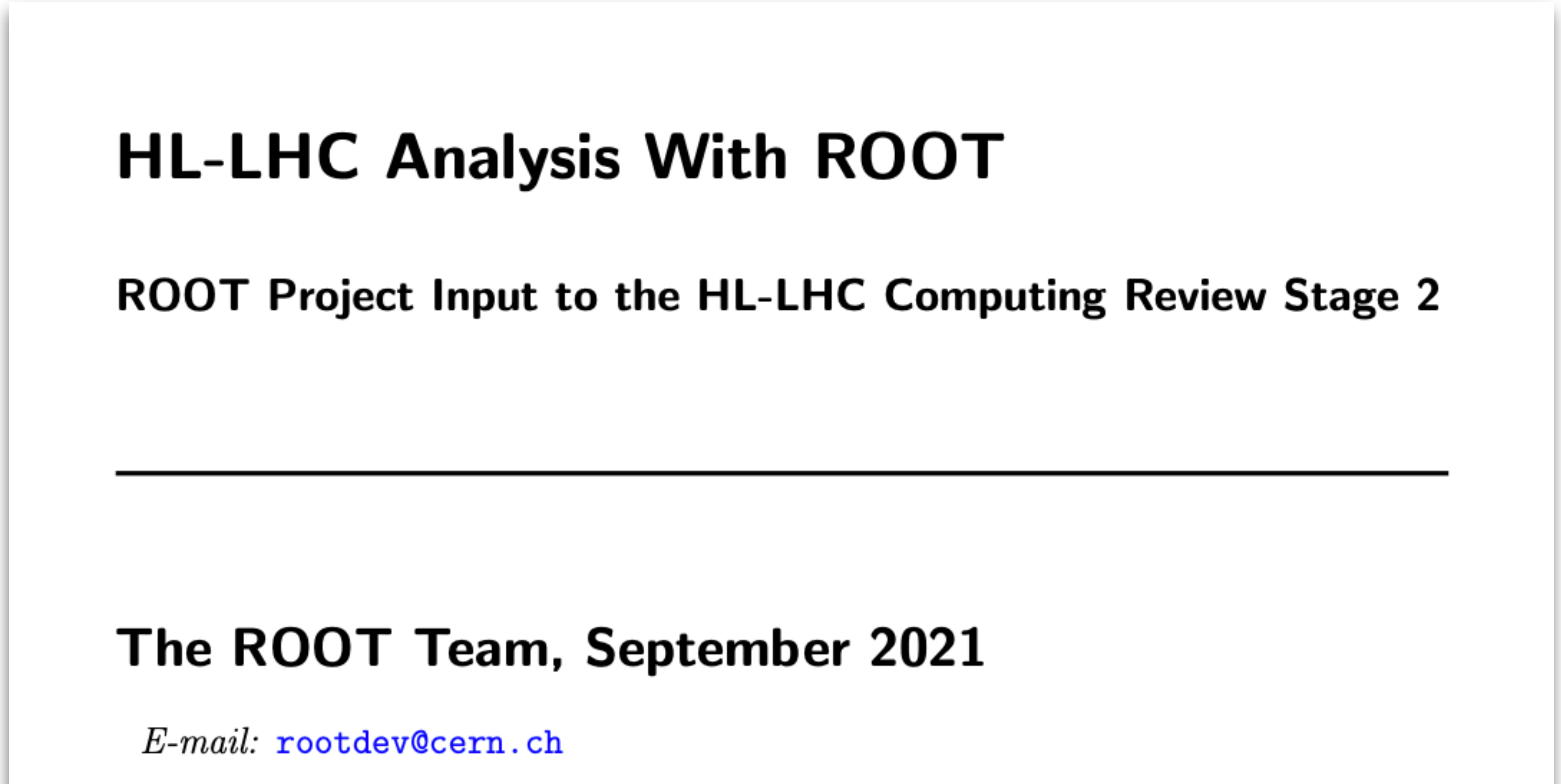
- Member of CERN's Open Science working group, CERN-IT Analysis Facility working group

Trainings

- Contributions to CMS Data Analysis school
- Software carpentries
- C++ course
- CERN Academic Training (SWAN)

LHCC Review

- Review of readiness for HL-LHC
- 60 pages of documentation of how ROOT works, what ROOT plans to do and why, risks and benefits
  - Significant load next to everything else
- One-day, hybrid event with reviewers: extremely constructive and helpful
- Report expected for March



HL-LHC Analysis With ROOT

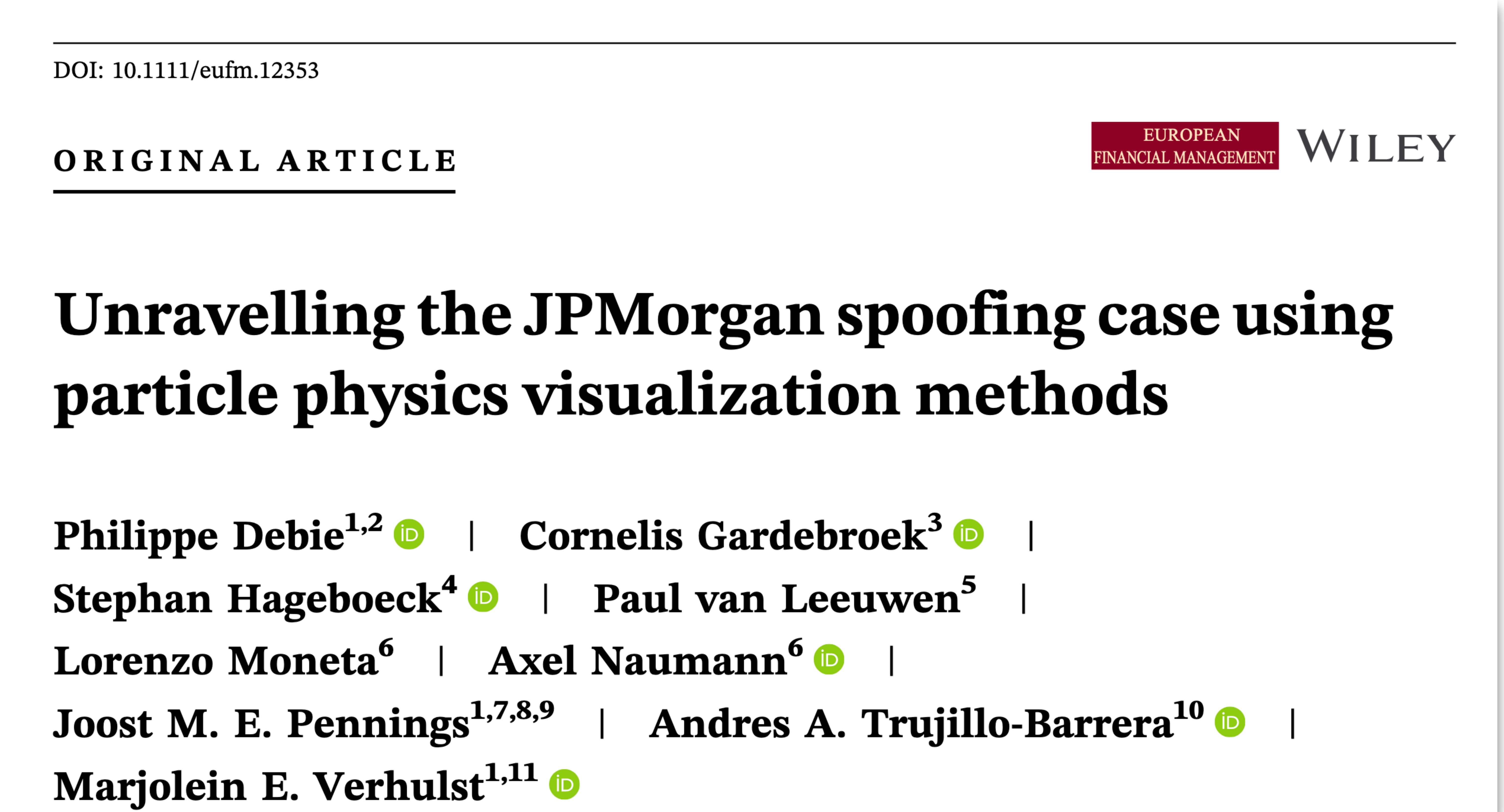
ROOT Project Input to the HL-LHC Computing Review Stage 2

The ROOT Team, September 2021

E-mail: rootdev@cern.ch

HighLO

- Cooperation with finance research on fraud detection
- Two publications in finance journals



DOI: 10.1111/eufm.12353

ORIGINAL ARTICLE

EUROPEAN FINANCIAL MANAGEMENT WILEY

Unravelling the JPMorgan spoofing case using particle physics visualization methods

Philippe Debie[1,2] | Cornelis Gardebroek[3] | Stephan Hageboeck[4] | Paul van Leeuwen[5] | Lorenzo Moneta[6] | Axel Naumann[6] | Joost M. E. Pennings[1,7,8,9] | Andres A. Trujillo-Barrera[10] | Marjolein E. Verhulst[1,11]

The Future of ROOT, the Future of Data Analysis

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

ROOT's Plans

- Unbeatable throughput for piping data into machine learning frameworks, training and inference

- 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"?

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

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

- 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

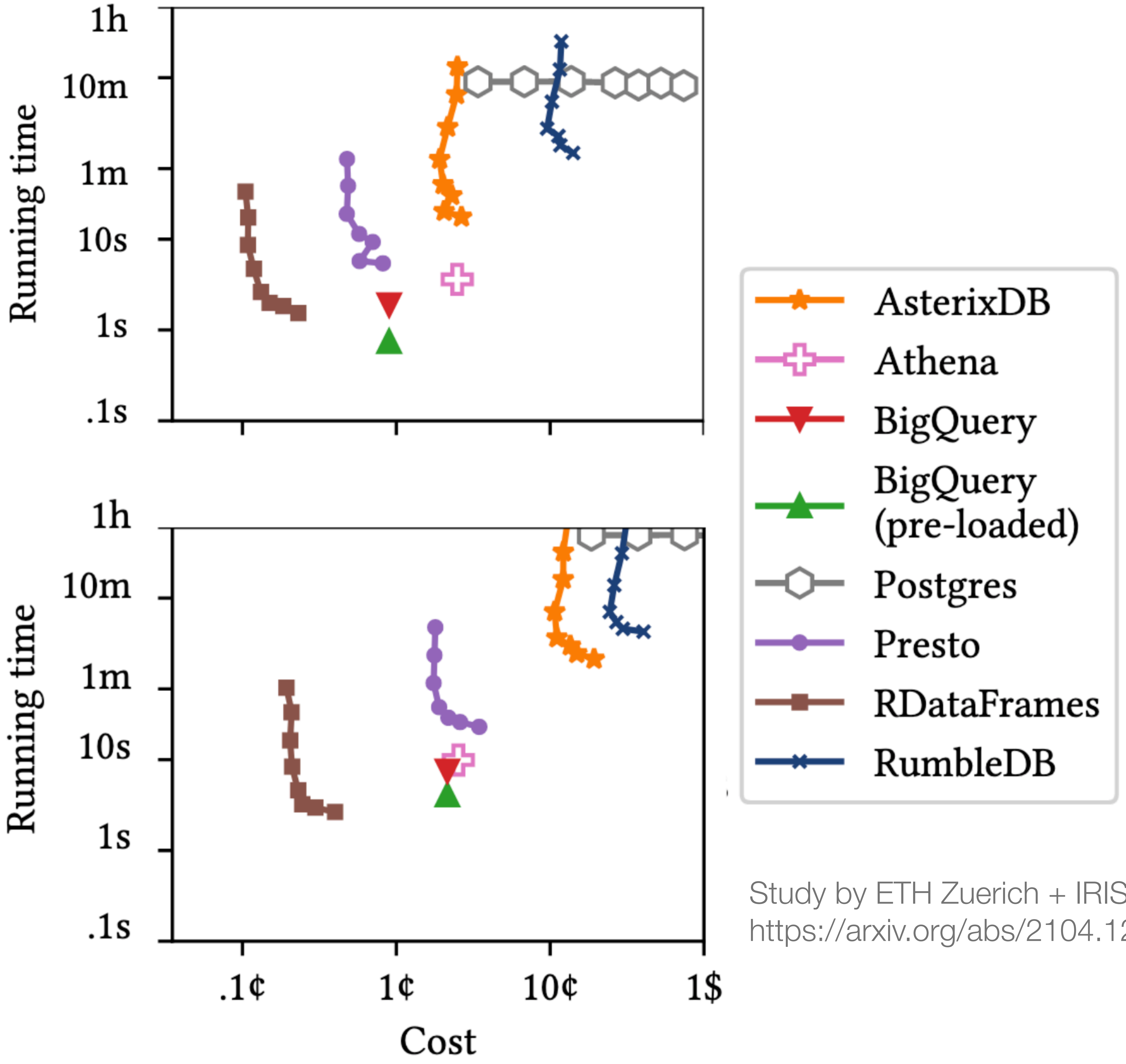
- Continuing separation of expert-level ROOT implementation, from user-level ROOT interfaces: harder to contribute for physicists, harder to "make it yours". ROOT becomes like Linux (install and use), instead of "a tool by us, for us"

- 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

Conclusion

Why?

Independent study shows ROOT's analysis interface RDataFrame to be significantly **faster**



Study by ETH Zuerich + IRIS-HEP
https://arxiv.org/abs/2104.12615

Goal

- ROOT provides an analysis interface that's
  - Reliable
  - Sustainable
  - Supported
  - Efficient
  - Simple

This Workshop

- Back at Fermilab after 21 years!
- Presentations by the ROOT team with all the details
- Presentations by users, providing feedback and criticism
- Discussions to understand, underline, convince, spark new ideas
  - *THE* crucial part of this workshop!
  - Zoom (raise hand), Zoom chat, or https://root-forum.cern.ch

THANK YOU

for caring about ROOT!

Martin Vorel, CC BY-SA 4.0, via Wikimedia Commons