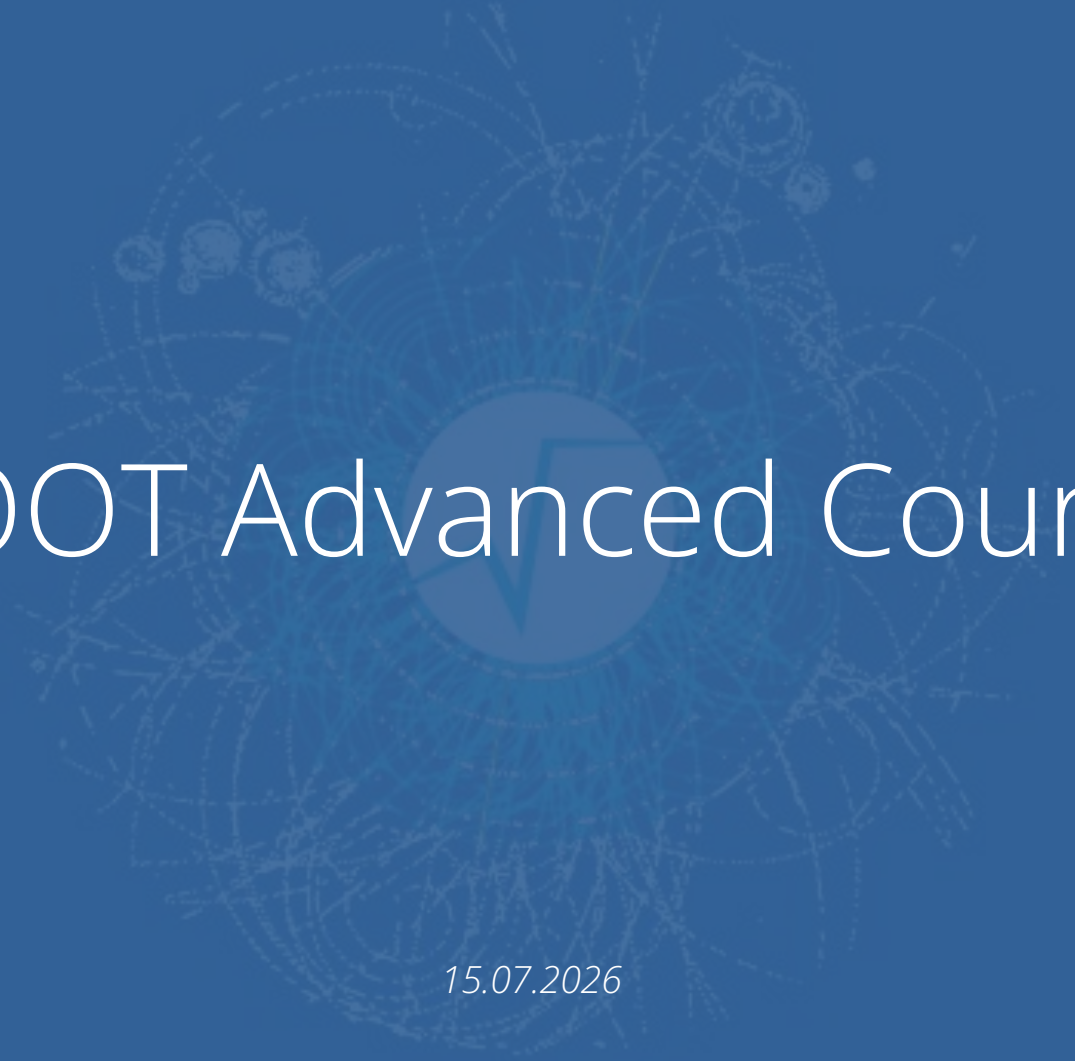




ROOT Advanced Course

15.07.2026



- Go to <https://github.com/root-project/ROOTAdvancedCourse>
- If you have CERN Computing account → click on the SWAN badge

ROOT Advanced Course





Use the Software Stack LCG 110

Found under “Other releases”

Leave all other parameters as-is!

Configure your session

[Generate URL](#)

Software

Try out our new experimental interface based on JupyterLab and let us know your feedback!

User interface [more...](#)

☐ Try the new JupyterLab interface (experimental)

Source [more...](#)

☒ LCG

☐ Container Environment

Software stack [more...](#)

AlmaLinux 9 (gcc13)

Platform [more...](#)

AlmaLinux 9 (gcc13)

Environment script [more...](#)

e.g. \$CERNBOX_HOME/MySWAN/myscript.sh

Session resources

CPU [more...](#)

2

Memory [more...](#)

8 GB

GPU [more...](#)

None

External computing resources

Spark cluster [more...](#)

None

HTCondor pool [more...](#)

None

[Start new session](#)



Setup - no CERN account

- Go to <https://github.com/root-project/ROOTAdvancedCourse>
- If you don't have CERN Computing account:
 - Click on either GH Codespaces or binder badges - note binder might take a bit of time to load
 - If you choose GH Codespaces - use Python venv as environment

ROOT Advanced Course





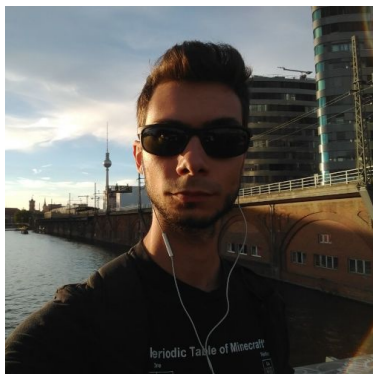
Meet the team



Silia



Florine



Giacomo



Jonas



- Run your own analysis leveraging the tools introduced today
- In particular – perform your analysis with RDataFrame

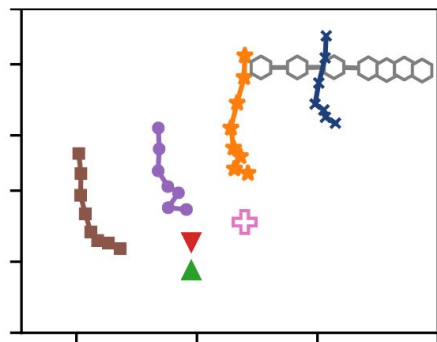


Why RDataFrame?

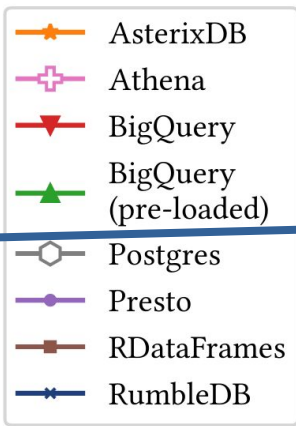
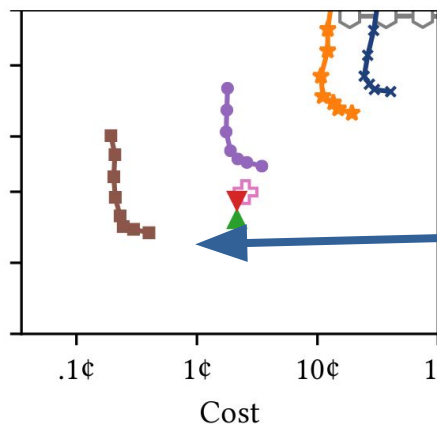
"...the general-purpose

data processing systems are significantly less performant than the domain-specific ROOT framework—due to limited scalability and inefficient handling of the data and queries relevant to HEP"

D. Graur, I. Muller, M. Proffit, G. Fourny, G. Watts et al,
[Evaluating Query Languages and Systems for High-Energy Physics Data](#), April 2021



(d) (Q4)



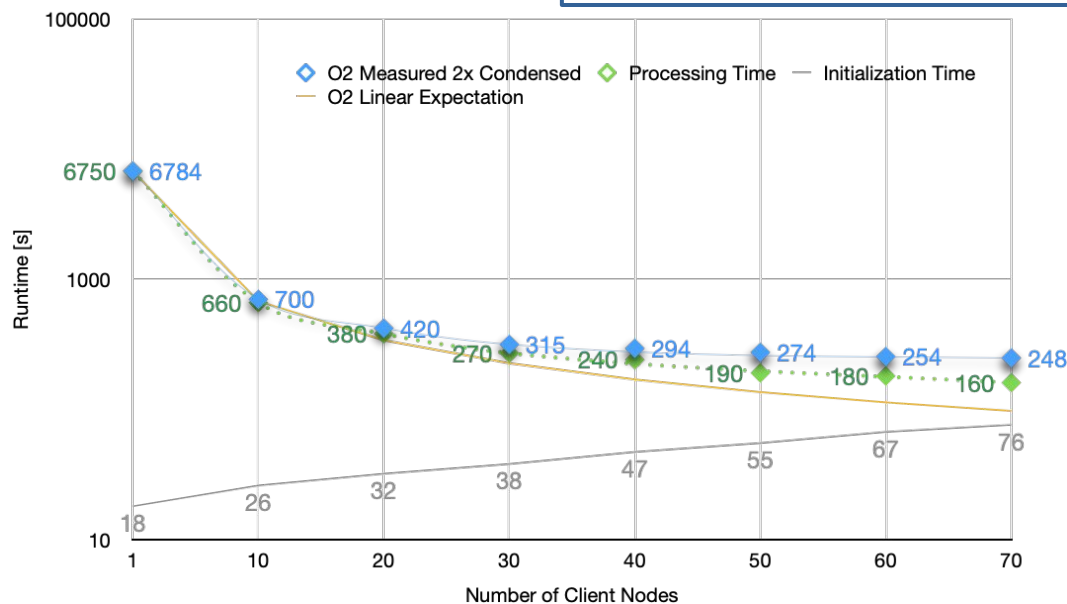
RDF: 10x more cost efficient
(note: this is for TTree, for
RNTuple even better)



Why RDataFrame?

A.J. Peters, J. Blomer
RNTuple Plenary
CHEP24
October 2024

Single Analysis
reaches avg. INGRES
222 GBit/s
during processing
345 GBit/s

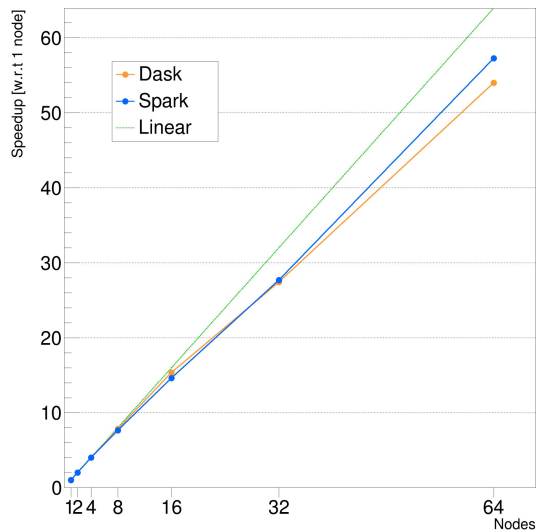




CERN HPC

Slurm jobs (Spark/Dask)
~100 GB/s on 2048 cores

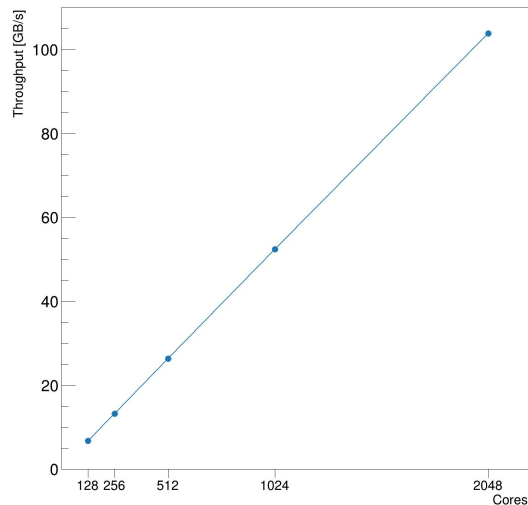
V.E. Padulano et al,
[Leveraging State-of-the-Art
Engines for Large-Scale Data
Analysis in High Energy Physics](#),
February 2023



Jülich HPC

Collaboration with OpenLab
Slurm jobs (via Dask)

J. Boulis,
[Benchmarking Distributed
Analysis at the Jülich HPC
Center](#), August 2023



DataFrame?



Why RDataFrame?



Finanziato
dall'Unione europea
NextGenerationEU



Ministero
dell'Università
e della Ricerca

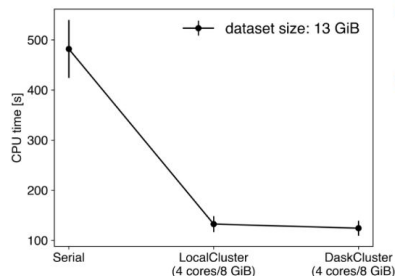


Italiadomani
PIANO NAZIONALE
DI RIPIEGLIA E RESILIENZA



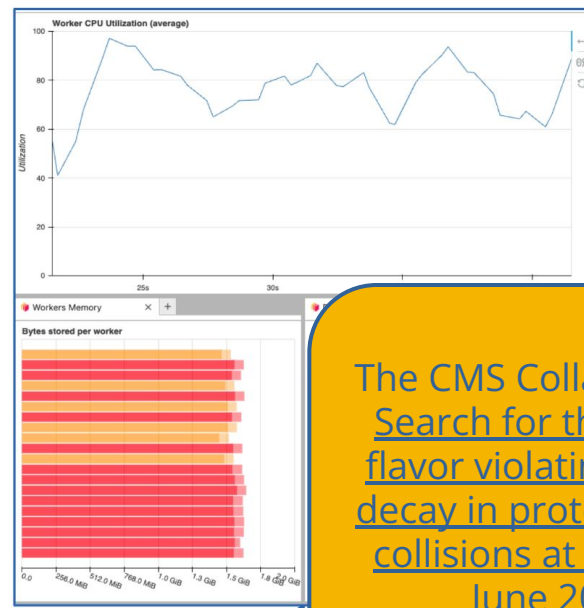
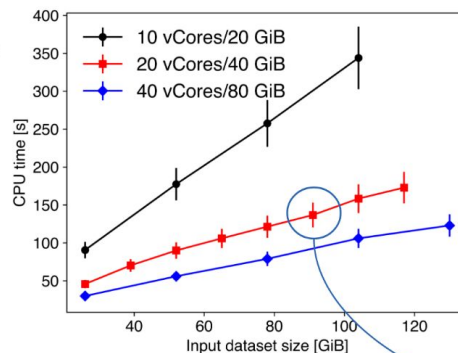
Centro Nazionale di Ricerca in SPDC,
Big Data and Quantum Computing

Preliminary results



- Stress test at high CPU and memory occupancy
- Stable performance, linearly scaling with the input dataset size
- Dataset size ~ 100 GiB is representative of ~15 /fb of Run3 data for this specific analysis

- Significant improvement in execution time *wrt* the standard/serial approach
- The facility allows for dynamically scaling the resources, here testing the performance at fixed #cores and memory, varying the dataset size



The CMS Collaboration,
Search for the lepton
flavor violating $\tau \rightarrow 3\mu$
decay in proton-proton
collisions at $\sqrt{13}$ TeV,
June 2024



G. Petrucciani, [Feedback from CMGRDF and distributed processing](#),
December 2024

The analysis used for these tests

- A complete CMS analysis on run 2 + 2022 data, at full complexity
 - [correctionlib](#) + [CMSJMECalculator](#) for applying scale factors, calibrations, uncertainties, ...
 - Onnx C++ runtime to evaluate DNNs from RDataFrame
- Measurements in a rare final state (top + Z processes), so only a modest total number of events after preselection, but from many different datasets
 - Often needing different computation graphs depending on data taking era, MC sample, ...

Dataset	Graphs (unique)	JITed functions	Files	Evts	Size
Just 2018	70 (43)	138	63	66 M	214 GB
Full dataset (2016-2022)	412 (248) (x 5.8)	260	362 (x 5.7)	273 M (x 4.1)	977 GB (x 4.6)



sics
3
24



Why RDataFrame?

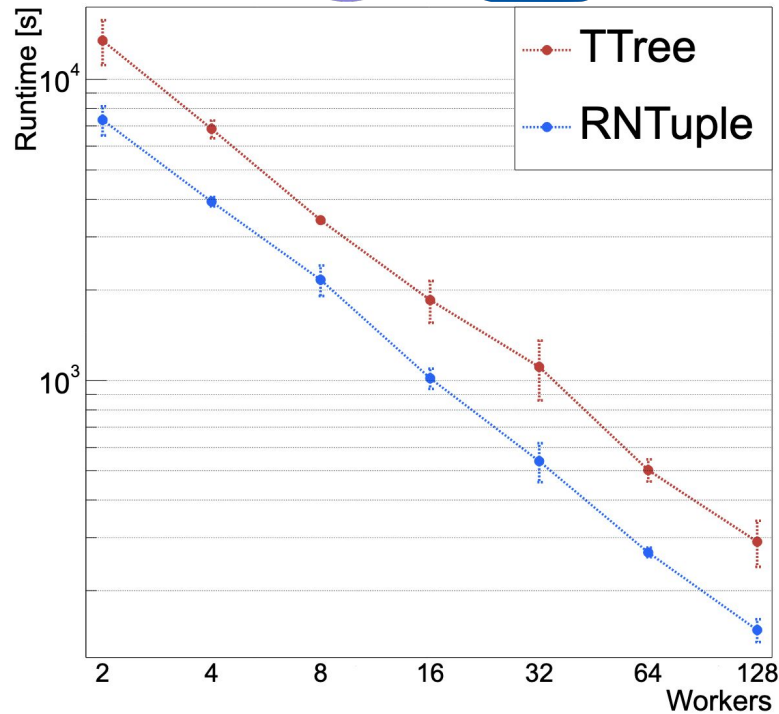
The Analysis Grand Challenge

RDF implementation:

collaboration between ROOT and
IRIS-HEP [\[1\]](#), [\[2\]](#)

Linear scaling on SWAN
RNTuple always faster

M. Czurylo et al, [Distributed
analysis in production with
RDataFrame](#), October 2024



1 2 3 4 5 6

16



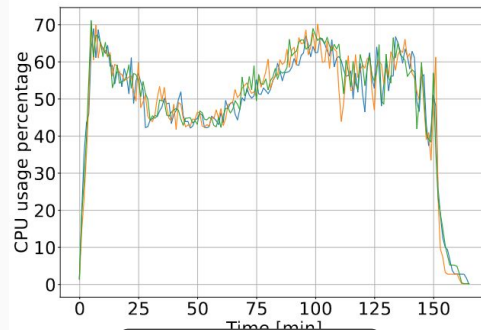
Why RDataFrame?

CMS analysis used on
Analysis Facilities

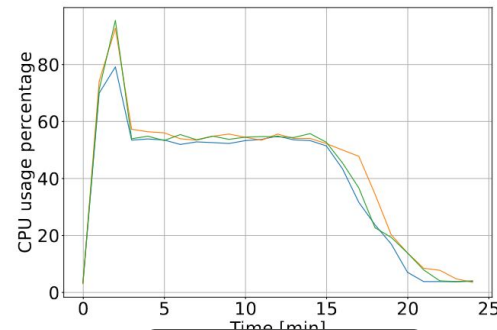
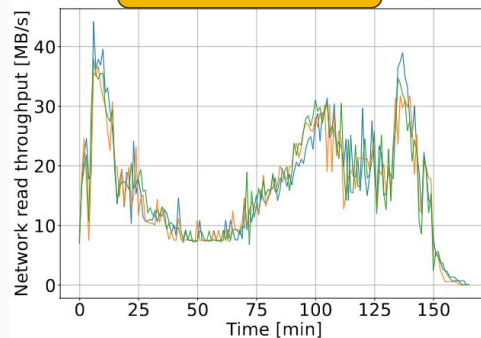
Before: Python for-loop with
NanoAODtools, manual job
submission

After: Interactive distributed
RDataFrame $\rightarrow$ O(10)
speedup

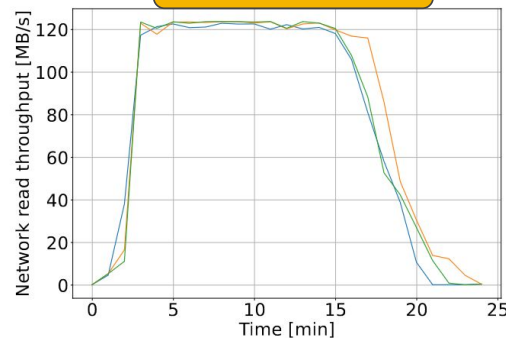
T. Tedeschi et al. Prototyping
a ROOT-based distributed
analysis workflow for HL-LHC:
The CMS use case, February
2024



Legacy



Distributed RDF



2 4 8 16 32 64 128
Workers



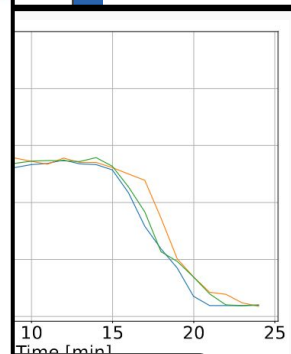
Status of my tests at various AF

	subMIT (MIT)	SWAN (CERN)	EAF (Fermilab)	Purdue	coffea casa (UNL)
access	MIT account	SSO cern	account + VPN	SSO cern	SSO cern
software stack	CONDA myenvAF	'Bleeding Edge' 'lcg/views/LCG_10 5a_swan'	CONDA client Singularity on workers	CONDA myenvAF	CONDA myenvAF
xcache		not planned	not planned		
interactive Local Cluster (one node)					
Distributed (multi node)	DASKGateway+ SLURM	Spark + HTcondor	DASKGateway + HTCondor	DASKGateway + K8s	DASKGateway + K8s
	DASK + SLURM	DASK + HTcondor			

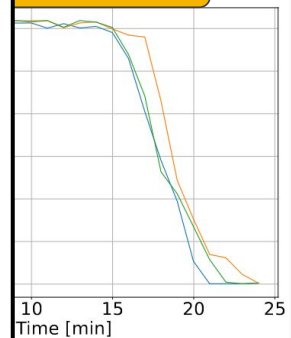
- in progress
- work
- didn't work
- not tried

M.D'Alfonso (MIT)

23



Distributed RDF



128
ers



J. Bendavid, CMS W Mass analysis with RDataFrame,
October 2024

Why RDataFrame?

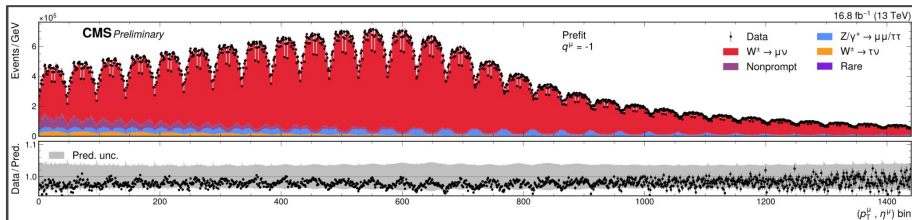
Summary

Increasing amount of data opens new opportunities

- Software developments must be ahead to fully exploit potential

Fast analysis turnaround was essential for this complex measurement

- RDF provides a convenient and efficient library
 - Initially showstoppers observed in scaling
 - Extensive work on critical parts to improve RDF and histogram implementation
- Full analysis runs in ~hours



ROOT RDataFrame

RDataFrame

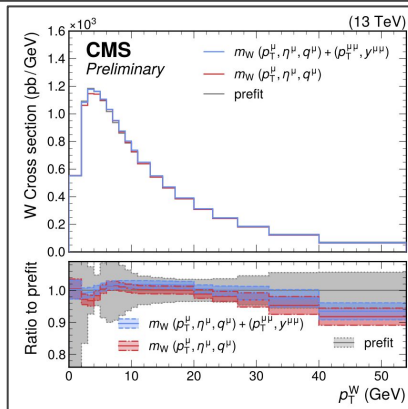
~1-2 h

Boost
Histograms

Critical parts in c++

- Functions: e.g. check if reco muon has a match to any gen muon

```
bool hasMatchDR2(const float& eta, const float& phi, const Vec_f& vec_eta, const Vec_f& vec_phi, const float dr2 = 0.09) {
    for (unsigned int jvec = 0; jvec < vec_eta.size(); ++jvec) {
        if (deltaR2(eta, phi, vec_eta[jvec], vec_phi[jvec]) < dr2) return true;
    }
    return false;
}
```



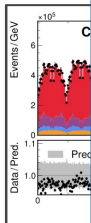


Why RDataFrame?

J. Gooding - RDataFrame in LHCb Offline Selection, October 2023

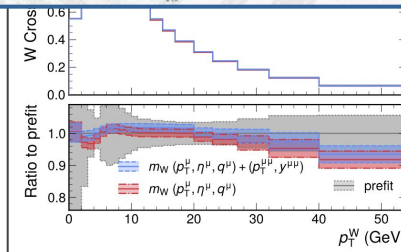


- ▶ RDataFrame provides **notable speedup** of offline selection, *if implemented optimally*:
 - ▶ Potential speedup of $\mathcal{O}(10)$ in some cases.
- ▶ uproot+numba **matches RDataFrame performance** in many scenarios:
 - ▶ I/O bottleneck identified → performance improvement possible.
- ▶ Both tools suitable for **fast parallelised selection**, necessary for upcoming Run 3 analyses.



- Functions: e.g. check if reco muon has a match to any gen muon

```
bool hasMatchDR2(const float& eta, const float& phi, const Vec_f& vec_eta, const Vec_f& vec_phi, const float dr2 = 0.09) {  
  
    for (unsigned int jvec = 0; jvec < vec_eta.size(); ++jvec) {  
        if (deltaR2(eta, phi, vec_eta[jvec], vec_phi[jvec]) < dr2) return true;  
    }  
    return false;  
}
```





Why RDataFrame?

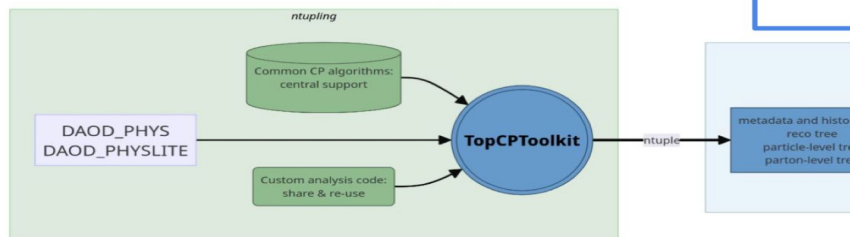
T. Dado, [ATLAS FastFrames](#), February 2025

FastFrames performance

- **Tested with a realistic example**
 - TTbar single lepton sample with ~ **20M events** (200M events in the DAOD)
 - ~ 400 GBs
 - Producing many control histograms
 - Producing analysis specific histograms
 - Running on all (automatic) systematics: ~400 systematics
 - Total of >25k histograms (total in three regions)
- **Ran on Chicago analysis facility**
 - systematics run, **25k+ histograms**, 10 GB ram requested, **10 CPUs**, running time: **10h 12m**
 - systematics run, **25k+ histograms**, 20 GB ram requested, **50 CPUs**, running time: **2h 52m**
 - systematics run ~**20k histograms**, 20 GB ram requested, **50 CPUs**, running time: **2h 20m**
 - nominal only run, **60 histograms**, 20 GB ram requested, **50 CPUs**, running time: **< 2m**
- **HTCondor**
 - nominal only run, ~**4k histograms**, ~**250 regions**, **30 CPUs**, running time: **< 16 m**
 - ~ **66M events**

Introduction

• What is FastFrames (FF) ?



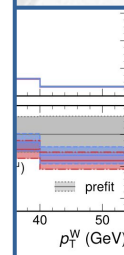
- Is a framework to produce **histograms/ntuples**.
- Compatible with the outputs produced by frameworks using **ATLAS format**
- Build around **ROOT::RDataFrame** technology.

• What can FF do ?

- Run in Linux and MacOS.
- NTuples → **NTuples**.
- NTuples → **Histograms**.
 - Supports 1D, 2D, 3D and TProfile
- Process **reco and truth trees** – including event matching.
- Prepare inputs for **unfolding** – which can be passed to [TRExFitter](#) directly.

5 analyses.

26th October 2023 | 12





- First part of the course ~ 1h
 - Notebooks 1 and 2 (I/O and RNTuple, UHI)
- Break ~ 30mins
- Second part of the course ~ 1h
 - Notebooks 3, 4 and 5 (RDataFrame, RDataLoader, RooFit)
 - Real analysis example
- Wrap up + time for Q&A ~ 30mins



We now move to the course material on Jupyter Notebooks



A look towards the future of ROOT



Improved Python packaging

- [Conda](#): all ROOT releases are published on conda-forge, for Linux and MacOS.

```
conda install conda-forge::root
```

- ROOT is published on PyPI, since version 6.38.00
 - Still in alpha mode, not ready for production
 - Try it out and let us know!

```
pip install ROOT
```



Jonas Hahnfeld, ROOT
PPP188

- Modern histogram package for ROOT 7
 - Well tested and high performance
 - Can be built standalone, without the rest of ROOT
- Key features compared to TH*
 - User-defined bin content types → template class
 - Still not go overboard with templates (Boost.Histogram, old prototype)
 - Arbitrary dimensions with dynamic axis types at run-time
 - See presentation at previous PPP: <https://indico.cern.ch/event/1403741/>
 - Efficient concurrent filling (memory and performance)
 - Histograms as data structures, no drawing and fitting



Jonas Hahnfeld, ROOT
PPP188

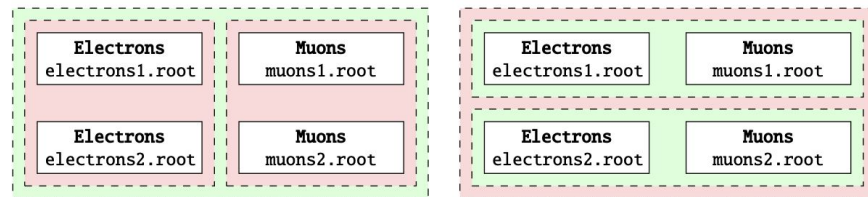
- `RHistEngine::FillAtomic`
 - Low contention without global statistics
 - No change of bin content type!
 - Keep full performance of sequential filling
- Concurrent filling of `RHist`:
 - Wrap histogram in `RHistConcurrentFiller`
 - Create `RHistFillContexts` that efficiently accumulate `RHistStats`

Check out the tutorials at
https://root.cern/doc/v638/group_tutorial_histv7.html



RNTuple tooling: RNTupleProcessor

- Event iteration API
- Enabling dataset compositions (chains and joins)
- An evolution of TTree friends, fit for more complex analysis requirements
- Tutorials available [\[1\]](#), [\[2\]](#)



[F. de Geus, CHEP'24](#)



- A new ROOT file interface
- No changes to on-disk format
- Modern, minimal interface
- Find tutorials [online](#)

```
# Reading
with RFile.Open("file.root") as file:
    # From top-level dir:
    hist = file.Get("hist")

    # From sub dir:
    subhist = file.Get("my/path/hist")

# Writing
with RFile.Recreate("file.root") as file:
    # To top-level dir:
    file.Put("hist", hist)

    # To sub dir:
    file.Put("my/path/hist", hist)

# Querying
for key in file.ListKeys("my"):
    print(key.GetPath())
```



Wrap up



Other examples of Analyses with RDF

- RDataFrame implementation of Analysis Grand Challenge (CMS Open Data)
<https://github.com/root-project/analysis-grand-challenge>
- LHCb example:
<https://github.com/siliataider/root-lhcb-open-data-analysis>



- ROOT web site: **THE** source of information for ROOT users
 - For beginners and experts
 - Installation instructions
 - Class documentation
 - Learn section
 - Access to the forum
 - Blog about recent developments

The screenshot shows the ROOT Data Analysis Framework website. The header includes the ROOT logo and navigation links: About, Learn, Install, Forum & Help, Manual, Blog Posts, Contribute, and For Developers. The main banner features the text "ROOT: analyzing petabytes of data, scientifically." and "An open-source data analysis framework used by high energy physics and others." Below this are buttons for "Learn more" and "Install v6.36.00". A row of icons represents different sections: Learn (graduation cap), Reference (open book), Forum (speech bubble), and Gallery (line graph). Below this, three featured articles are shown: "Enables processing and scientific analysis of large amounts of data: today, more than 2 exabytes are stored in ROOT files. The Higgs was discovered with ROOT!" (with a bar chart icon), "Open source, which means that you can use it freely and modify it. It adopts an open development process, inviting its users to contribute to it." (with a code icon), and "Delivered with a C++ interpreter, ideal for fast prototyping. ROOT also offers a Python interface to all its components with dynamic bindings. Jupyter notebooks are supported too." (with a dollar sign icon). At the bottom, there are sections for "Level Up Your Histogram Analysis Using ROOT's UHI" (dated 06 Jun 2025), "Make ROOT your own on macOS with Nix" (dated 01 May 2025), "ROOT Users Workshop 2025" (dated 14 Apr 2025), and "Latest Releases" with a list of versions and dates.



- ROOT Website: <https://root.cern>
- ROOT Forum: <https://root-forum.cern.ch>
- Materials for learning more: [Learn - ROOT](#)
- Documentation: <https://root.cern/doc/master/>
- Github: <https://github.com/root-project/root>