

RNTuple Workshop 2024: State of Affairs

Jakob Blomer for the ROOT team
2024-12-02

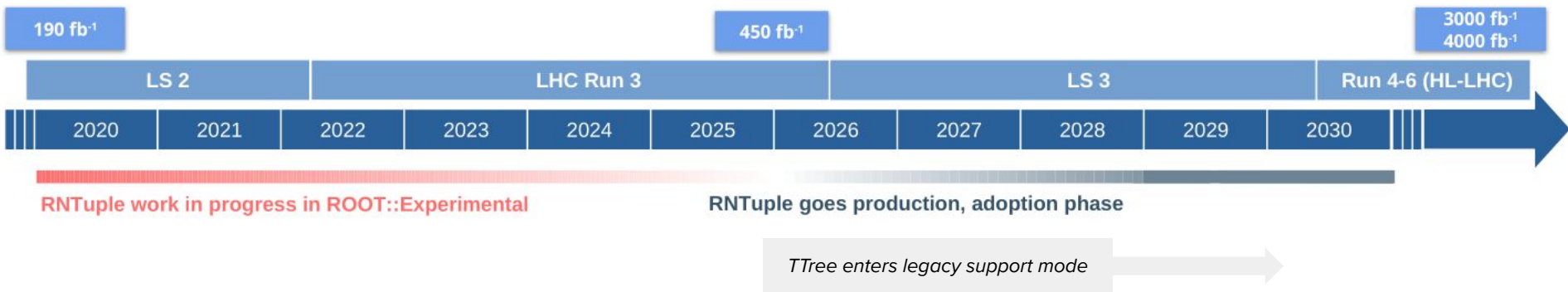
ROOT

Data Analysis Framework

<https://root.cern>



Context: ROOT I/O Upgrade for HL-LHC



>2EB (now) → >10EB (end of HL-LHC)
~½ of the currently projected WLCG budget on storage

Major I/O upgrade of the event data file format and access API: TTree → RNTuple



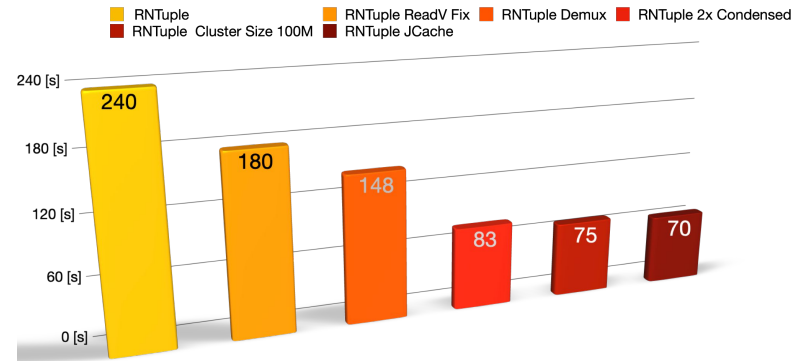
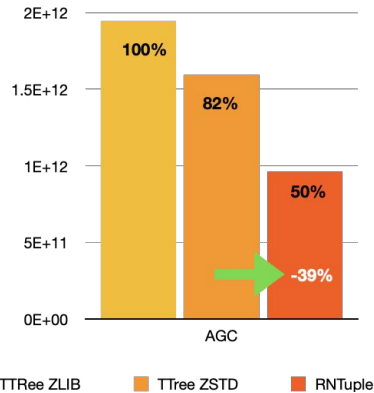
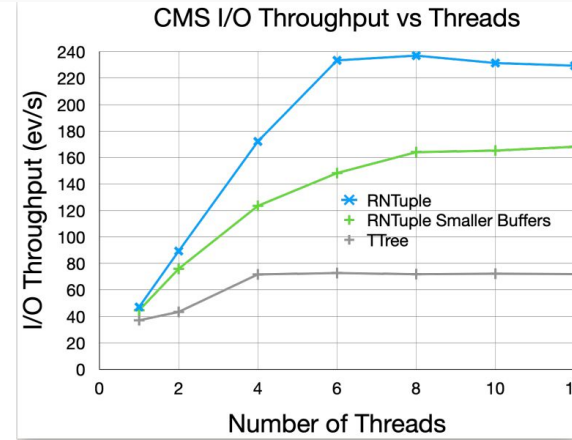
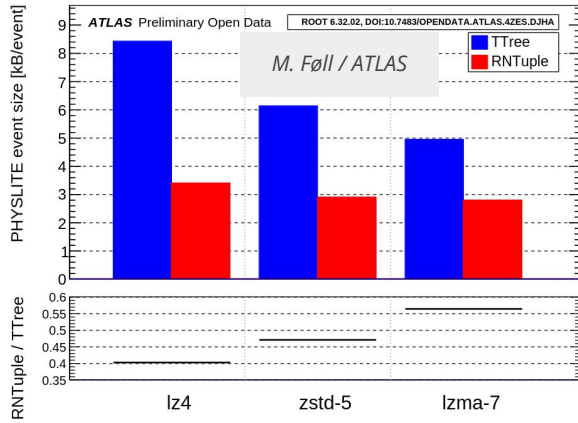
RNTuple Main Results

- **Major I/O upgrade** of the event data file format and access API: **TTree → RNTuple**
 - Less disk and CPU usage for same data content
 - **10-50% smaller** files, **better single-core performance often by factors**
 - Give access to **novel and future storage technologies**
 - Native support for HPC and cloud object stores
 - Async and parallel I/O: fully exploits modern NVMe drives
 - Design prepared for accelerators (e.g., GPUs, compression offloading)
 - Systematic use of checksumming and exceptions to prevent silent I/O errors
- Initial support in **ATLAS, CMS, LHCb software frameworks** (ESD, AOD, derived AODs & ntuples)
- Large-scale testing with IT storage group
 - 70 nodes, 100GbE EOS connection, 100TB inflated AGC benchmark
- ROOT 6.34 (Nov 2024): RNTuple stable on-disk format (version 1.0) released
 - Future ROOT versions will read data written with 6.34
 - Planned optional and possibly forward-compatibility breaking changes foreseen
- ROOT v6.36 (planned for Q2/2025): first set of APIs move out of ROOT::Experimental
 - Taking into account the input received by the HEP-CCE review

Many results presented at CHEP'24



Highlights from CHEP





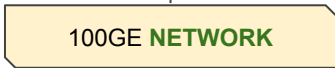
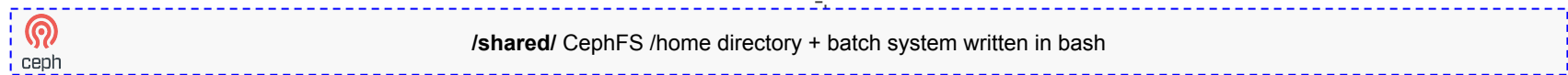
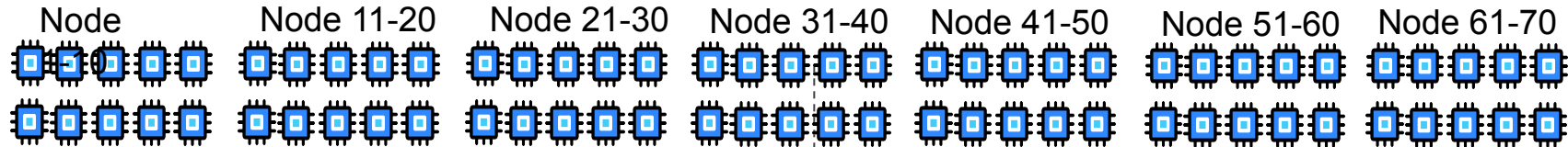
Progress since last year: type system

Type Class	Types	EDM Coverage	RNTuple Status
PoD	<code>bool, char, std::byte, (u)int[8,16,32,64]_t, float, double</code>	Flat n-tuple	Available
Records	Manually built structs of PoDs		
(Nested) vectors	<code>std::vector, RVec, std::array, C-style fixed-size arrays</code>	Reduced AOD	Available
String	<code>std::string</code>	Full AOD / ESD / RECO	Available
User-defined classes	Non-cyclic classes with dictionaries		Available
User-defined enums	Scoped / unscoped enums with dictionaries		Available
User-defined collections	Non-associative collection proxy	Full AOD / ESD / RECO	Available
stdlib types	<code>std::pair, std::tuple, std::bitset, std::(unordered_) (multi)set, std::(unordered_) (multi)map</code>		Available
Alternating types	<code>std::variant, std::unique_ptr, std::optional</code>		Available
Streamer I/O	All ROOT streamable objects (stored as byte array)	Full AOD / ESD / RECO	Available
Low-precision floating points	<code>Double32_t, f16</code>		Available
	Custom precision / range (bfloat16, TensorFloat-32, other AI formats)	Optimization benefitting all EDMs	Available



Progress since last year: AGC Testing I


COMPUTE



Max read 40 GB/s

Price per Volume $\$ \frac{1}{x}$ EOS Relative Price $\$ 4$


EOSPILOT
14 nodes **100GE** 1334x 18TB **HDDs**
24 PB - 20 PB usable



380 GB/s

$\$ \frac{1}{x}$ EOS $\$ 35$

EOSALICEO²
125 nodes **100GE** 12000x **HDDs**
180 PB - 150 PB usable



22.5 GB/s

$\$ \frac{6}{12x}$ ceph $\$ 0.8$

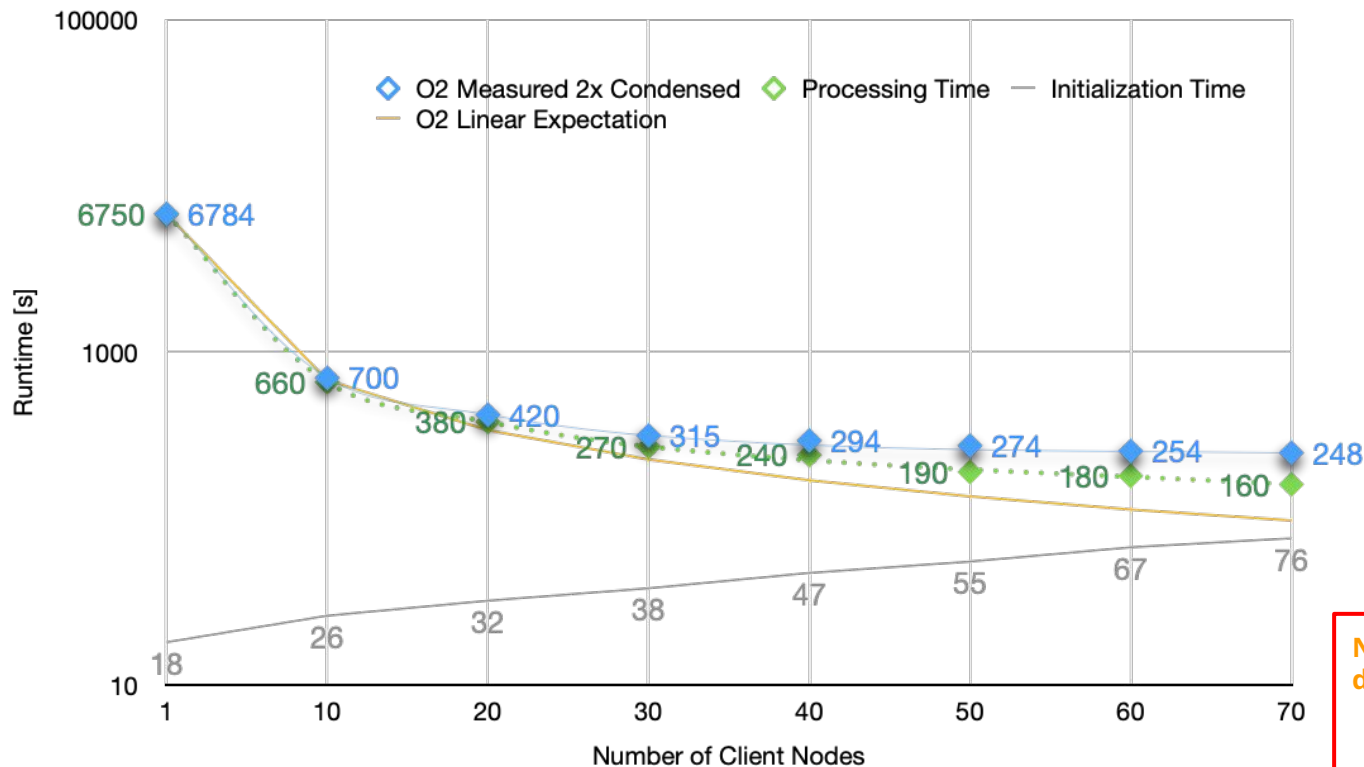
CephFS
8 nodes **25GE**
80 x 7.6 TB **NVMe**
568 TB - 284 TB usable

STORAGE



Progress since last year: AGC Testing II

- Introducing modified RNTuple format for **AGC²⁰⁰** with **EOSALICE²**



With a 100x inflated **AGC²⁰⁰** dataset we observe that as the number of client nodes increases, the initialization time gets close the processing time, resulting in a breakdown of scalability.

Single Analysis

extremely sparse
reaches avg. INGRES
222 GBit/s
during processing
345 GBit/s

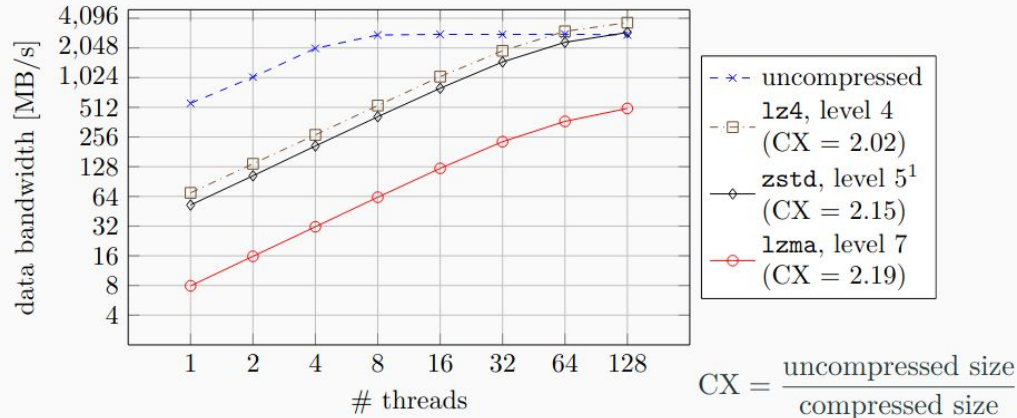
Next step: Reconstruction and/or data derivation benchmark(s)

- Dense reading and (parallel) writing

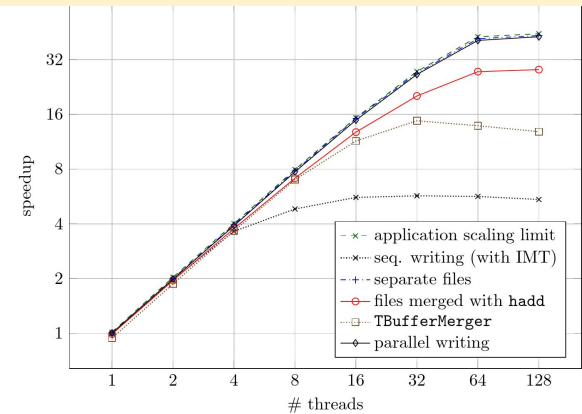


Progress since last year: parallel writing & direct I/O

Reconsider trade-off between write speed and file size



Writing in parallel to one file is as fast as writing 128 files



→ <https://indico.cern.ch/event/1338689/contributions/6010002>

→ <https://arxiv.org/abs/2410.14239>

- Truly parallel writing; prototype support for multi-process and MPI support
- Capable of fully exploiting NVMe drives
- Reaching throughput values that allow for meaningful contribution to processing workflow of DUNE supernova event candidates



- RNTupleProcessor: friends & chains with solid underpinnings
 - <https://indico.cern.ch/event/1338689/contributions/6016196>
 - See talk by Florine later today:
<https://indico.cern.ch/event/1468611/#3-rntuple-processor-joins>
- Connect RNTuple type description to TFile streamer info (enabling, e.g., MakeProject and manual schema evolution)
- Late model extension in RNTupleMerger (TFileMerger)
- Removal of 1GB TFile limit for RNTuple data (exception: streamer field)
- Tested limits: 100k columns, 100k clusters, 600M elements per page
 - Some factor of 10 larger than largest examples we encounter today (e.g., ~15k columns in CMS AOD)



Tooling: RNTupleViewer

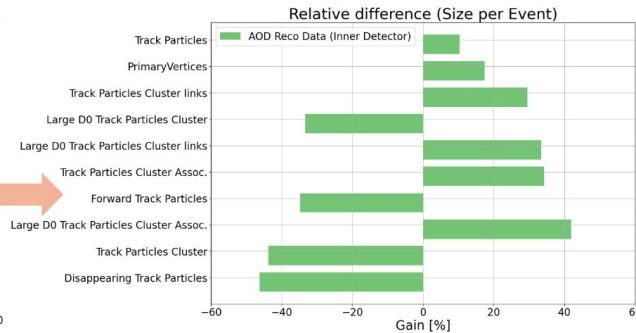
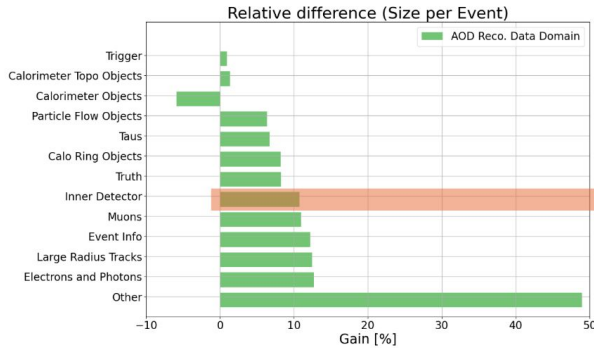
Rich (internal) tooling: RNTupleInspector (presented last year), RNTupleViewer: <https://codeberg.org/silverweed/rntviewer>

```
000: 72 6F 6F 74 80 80 F8 0D 00 00 00 64 80 00 09 D2 00 00 09 9F 00 00 00 33 00 00 00 01 00 00 00 3C
020: 04 00 00 00 00 00 00 00 12 00 00 01 0D 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
040: 00 00 00 00 5C 5C F8 9F 98 A3 2F 2F 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
060: 00 00 00 00 00 00 00 00 04 00 00 00 4A 76 DA D4 18 00 00 00 00 00 00 00 00 00 00 00 00 00
080: 46 69 6C 65 0C 52 4E 54 75 78 6C 65 2E 72 6F 6F 74 80 3C 5E 4E 54 75 78 6C 65 2E 72 6F 6F 74
0A0: 05 76 DA D4 18 76 DA D4 18 63 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0C0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0E0: 00 04 00 00 01 4C 76 DA D4 18 00 22 00 01 00 00 00 DC 00 00 00 64 05 52 42 6C 6F 62 00 00 01 00
100: 4C 01 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 43 6F 6E 74 72 69 62 75 74 6F 72 73 17 00
120: 00 00 54 68 65 20 66 69 72 73 74 20 65 76 65 72 20 52 4E 54 75 78 6C 65 2E 0E 0E 00 00 00 52 4E 4F
140: 54 20 76 36 2E 33 95 2E 30 30 31 7D FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
160: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 66 69 72 73 74 4E 61 6D 65 00 00 00 00
180: 73 74 64 3A 3A 73 74 72 69 6E 67 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
1A0: 00 00 00 01 00 00 00 00 00 00 00 00 00 00 00 00 6C 61 73 74 4E 61 6D 65 0B 00 00 00 73 74 64 3A 3A
1C0: 73 74 72 69 6E 67 00 00 00 00 00 00 00 00 00 00 A4 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
1E0: 00 00 0F 00 40 00 00 00 00 00 00 00 00 00 00 00 14 00 00 00 00 00 00 00 00 00 00 00 00 00 00
200: 00 00 14 00 00 00 00 00 00 00 0F 00 00 00 01 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
220: 00 00 01 00 00 00 00 00 00 00 F4 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
240: 00 00 70 CA FD 11 D2 6D B5 81 00 00 03 15 00 04 00 00 02 D3 76 DA D4 18 00 22 00 01 00 00 02 4A
260: 00 00 00 64 05 52 42 6C 6F 62 00 00 05 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
280: 00 00 00 00 17 00 00 00 00 00 00 00 1C 00 00 00 00 00 00 00 24 00 00 00 00 00 00 00 00 00 00
2A0: 00 00 00 00 2D 00 00 00 00 00 00 00 33 00 00 00 00 00 00 00 39 00 00 00 00 00 00 00 00 00 00
2C0: 00 00 00 00 46 00 00 00 00 00 00 00 57 00 00 00 00 00 00 00 67 00 00 00 00 00 00 00 00 00 00
2E0: 00 00 00 00 7C 00 00 00 00 00 00 00 8B 00 00 00 00 00 00 00 99 00 00 00 00 00 00 00 00 00 00
300: 00 00 00 00 9D 00 00 00 00 00 00 00 8A 00 00 00 00 00 00 00 B2 00 00 00 00 00 00 00 00 00 00
320: 13 AA 17 C0 4A 61 6B 6F 62 50 68 69 6C 69 70 70 65 41 78 65 6C 44 61 6E 69 6C 6F 53 69 6D 6F 6E
340: 42 65 72 74 72 61 6E 64 4D 61 78 4A 61 78 69 65 72 45 6E 72 69 63 6F 65 72 67 65 79 47 69 6F
360: 76 61 6E 65 61 4A 6A 72 72 79 46 6C 6F 72 69 65 65 20 57 69 6C 65 00 00 00 00 00 00 00 00 00 00
380: 61 72 64 20 4D 01 6E 65 72 65 64 56 69 6E 63 65 6E 7A 6F 20 45 64 75 61 67 62 64 6F 4A 6F 6C 6C 79
3A0: 41 6C 6E 65 74 74 69 6E 20 53 65 72 68 61 6E 4A 6F 6E 61 73 4D 61 63 69 65 6A 47 69 61 63 6F 6D
3C0: 6F 47 72 69 6F 6F 72 69 53 70 65 63 69 61 6C 20 74 68 61 6E 6B 73 81 86 FD 6F FD D5 88 34 06 00
3E0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 12 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
400: 00 00 00 00 00 00 29 00 00 00 00 00 00 2D 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
420: 00 00 00 00 00 00 44 00 00 00 00 00 00 52 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
440: 00 00 00 00 00 00 63 00 00 00 00 00 00 6B 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
460: 00 00 00 00 00 00 7B 00 00 00 00 00 00 84 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
480: 00 00 00 00 00 00 C1 00 00 00 00 00 00 3D D6 C1 AD 55 74 1B 48 42 6C 6F 6D 65 72 43 61 6E 61
4A0: 6C 4E 61 75 6D 61 6E 6E 50 69 70 61 72 6F 4C 65 69 73 69 62 61 63 68 42 65 6C 6C 65 6E 6F 74 4F
4C0: 72 6F 6B 4C 6F 70 65 7A 2D 47 6F 6D 65 7A 47 65 69 62 61 75 64 4C 69 6E 65 76 4C 61 7A 7A 61 72
4E0: 47 20 69 6F 74 6F 69 69 69 69 69 69 69 69 69 69 69 69 69 69 69 69 69 69 69 69 69 69 69 69 69
500: 6F 43 68 65 6E 4D 65 74 65 48 61 68 6E 65 65 6C 64 53 7A 79 6D 61 6E 73 6B 50 50 61 72 6F 6C 69
520: 6E 69 52 79 62 6B 69 6E 65 74 6F 20 61 6C 6D 20 6E 61 6D 65 77 6F 72 6B 20 64 65 76 65 6C 6F
540: 70 65 72 73 20 69 6E 20 74 68 65 20 65 78 70 65 72 69 6D 65 6E 74 73 62 64 7A 8D 95 54 39 0B 00
560: 00 01 16 00 04 80 00 00 F4 76 DA D4 18 00 22 00 01 00 00 00 00 00 00 00 64 05 52 42 6C 6F 62 00
580: 00 03 00 F4 00 00 00 00 70 CA FD 11 D2 6D B5 81 DC FF FF FF FF FF FF FF FF 01 18 00 00
5A0: 00 00 00 00 00 00 00 00 00 00 00 00 15 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
5C0: 00 54 FF FF FF FF FF FF FF FF 04 00 00 00 00 00 00 FF FF FF FF FF FF 01 EA FF FF FF FF 00 00 00
5E0: 00 6C 02 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
600: 00 4E FF FF FF B2 00 00 00 24 03 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
620: FF FF FF FF FF 01 00 00 EA FF FF FF FF 00 00 00 00 DE 03 00 00 00 00 00 00 00 00 00 00 00 00 00
640: 00 00 00 00 D8 FF FF FF FF FF FF FF FF 00 00 00 00 3F FF FF FF FF 00 00 00 00 96 84 00 00 00 00
660: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
680: DA D4 18 00 22 00 01 00 00 06 75 00 00 00 64 05 52 42 6C 6F 62 00 00 02 00 94 00 00 00 00 00 00
6A0: 00 00 00 00 00 00 79 CA FD 11 D2 6D B5 81 38 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
6C0: 00 00 00 F4 FF FF FF FF FF FF FF FF 00 00 00 00 00 00 F4 FF FF FF FF 00 00 00 00 F4 FF FF FF FF
6E0: FF FF FF 00 00 00 00 C4 FF FF FF FF FF FF FF FF 01 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
700: 00 00 00 16 00 00 00 00 00 00 00 01 00 00 00 F4 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
720: 00 00 00 01 70 73 0D 8E 1C 6E 7D 00 00 00 84 00 04 00 00 00 4E 76 DA D4 18 00 36 00 01 00 00 00 07
```

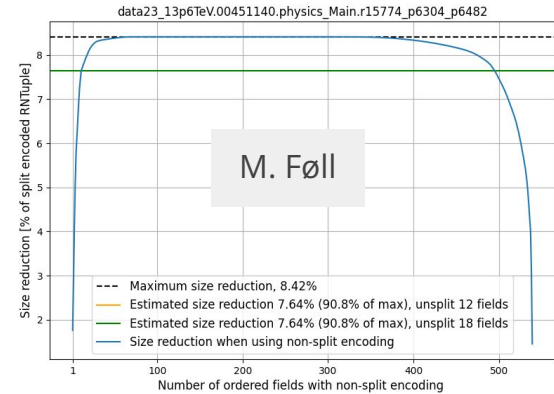
We can imagine a set of powerful tools, maintained outside the ROOT source tree.

E.g., manual RNTuple descriptor manipulation.

- Tuning (auto tuning?) of column encoding



→ <https://indico.cern.ch/event/1338689/contributions/6010824/>



- Investigation of MiniAOD space savings (~7.5 %, would ideally be > 10% [somewhat arbitrary])
- Framework support: profile & improve writing and reading from frameworks
- Support for vectors with custom allocators (ATLAS)
- Support for writing into directories, bulk reading optimizations (ALICE)
- Validation suite for 3rd party readers
- I/O support for SoA data structures, [see talk tomorrow](#)
- Meta-data support, [see talk tomorrow](#)



- Define the first set of APIs to move out of ROOT::Experimental
 - Planned for ROOT v6.36, i.e. likely May 2025
 - More or less the classes subject to the HEP-CCE review
 - We can extend the APIs later (e.g. additional ClusterPool tuning), but once in production it will be costly to change existing APIs
 - Not all RNTuple APIs will move out at the same time
- Fully functional schema evolution (basic functionality working for v6.36, full set possibly post v6.36)
- RNTupleProcessor: capability to arbitrarily combine friends and chains
- RNTuple attribute extension prototype (see later), likely leading to v1.1 ondisk format
- Testing and validation on IT testbed with data derivation and/or reconstruction benchmark(s)
- Tuning, support, bug fixes, training: with the transition to production, the support effort begins
- *Lower priority:* S3 backend, intra-event links, checkpoints during writing, sharded clusters and horizontal merge



Round table: questions to experiments

- Round table discussion starters:
 - From your point of view, are we missing anything important?
 - Is the parallel writer of interest to you?
 - How can we facilitate RNTuple adoption?
 - Can you provide us a benchmark (code + data) for a reconstruction or data derivation task for the IT testbed
 - Ideally: test also the parallel writer
 - Large data set would allow for validating possible automatic tuning of column encoding
 - Extra: we will (re)start development on RFile next year. What are your wishes (e.g. ownership model, concurrency, etc.)