RNTuple support and future plans

Andres Rios-Tascon

PRINCETON UNIVERSITY

iris hep

You either die a TTree, or live long enough to see yourself become an RNTuple

- RNTuple is the new serialization format that will replace TTree.
- It is much more modern, offers performance and storage improvements, and has a formal specification. Very similar memory layout to Awkward!
- Last year, the first stable version of the binary format was released. We can confidently write data with it, knowing it will continue to work in the future.
- Experiments have started the transition process.

RNTuples in Uproot

- Working towards a seamless transition from TTrees to RNTuples.
- Hopefully, users won't even realize anything changed. (Or only notice that things got better)

```
>>> import uproot
>>> with uproot.open("ntpl001_staff_rntuple_v1-0-0-0.root") as f:
...     rntuple = f["Staff"]
...     all_arrays = rntuple.arrays()
...     rntuple_field = rntuple["Age"]
...     rntuple_field_array = rntuple_field.array()
```

Updates regarding RNTuples in Uproot

- There is now 100% bidirectional support between RNTuple and Awkward.
- Almost complete API parity with TTree. Handling of many edge cases.
- RNTuples can be read directly to GPU with direct-storage.
- Dict-like assignment now writes an RNTuple instead of a TTree.
- RNTuples can be read into virtual arrays.
- Late-model extensions are now supported.
- Existing RNTuples can be extended.

} Fresh work from summer student

RNTuple support in other Scikit-HEP projects

UPROOT5-BROWSER

Fully supported!

Supported, but there are currently some performance issues. (Will work on it)

hep▸convert

Work not yet started, but should be straightforward.

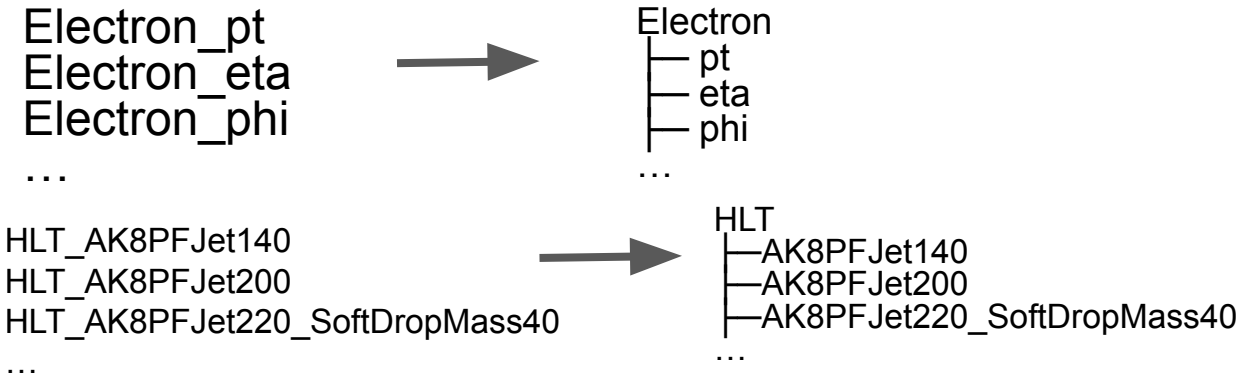
RNTuples in CMS

Slightly better situation compared to last year:

- CMSSW now has a new-enough version of ROOT to produce stable RNTuples.
- You can toggle RNTuple output for most workflows with a flag.
- As far as I can tell, RNTuples are still not ideal, since they are making heavy use of unsplit types.
- I’ve heard there have been some performance issues.
- The NanoAOD RNTuples continue to be mostly neglected.

Improving NanoAOD

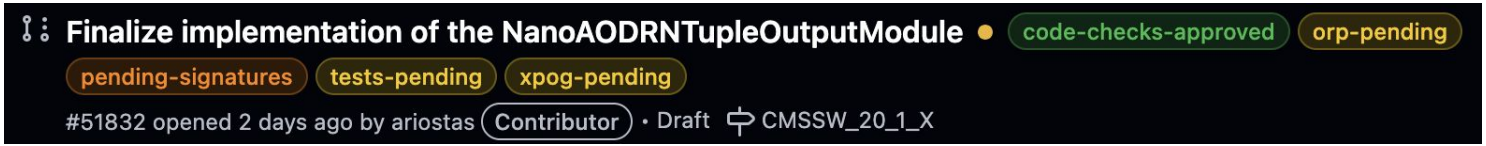
- The RNTuple format gives us the opportunity to organize things better.
- Last year I proposed a restructuring like this:



- The issue was that trigger flags in NanoAOD can be added midway through writing.

Extending the RNTuple spec

- I pointed out that the RNTuple spec didn’t explicitly disallow extending existing untyped records. (ROOT was actually able to read them just fine)
- I asked them to officially allow this in the spec, arguing for NanoAOD as a use case.
- They agreed and added an API for it.
- A recent special release of CMSSW added ROOT 6.40, so I’m finishing the restructuring.



RNTuples in other experiments

- In ATLAS: haven’t heard much, but it seems like the situation is similar to last year.
- No clue about other experiments.

Future plans for Uproot

Apart from the standard plans to improve performance and reliability, we have two big things planned for Uproot.

- Use Formulate for expression support.
- Use RootFileSpec for metadata parsing.

Switching to Formulate

- Uproot has a lot of machinery for parsing and evaluating expressions.
- I recently worked on reviving the Formulate package, which focuses on parsing and translating between NumExpr and ROOT expressions.
- We will use Formulate+NumExpr to add expression support for RNTuples, and eventually also replace the large chunk of code that handles expressions for TTrees.

Delegating metadata to RootFileSpec

- RootFileSpec (Nick Smith's package) is specifically designed to parse ROOT metadata, can handle more types than Uproot, and has a much cleaner design.
- Last summer there was a summer student working on this. There was some progress, but we couldn't finish a minimal integration.
- I still have to talk to Nick about it, since there were some potential plans to rewrite RootFileSpec in Rust.

Summary

- Uproot has fully transitioned from TTree to RNTuple.
- RNTuple support has very significantly improved since last year.
- Coffea and HEPConvert still need some work.
- NanoAOD RNTuples are being restructured.
- We plan to integrate Formulate and RootFileSpec with Uproot.

Thank you!