



CMS closeout on RNTuple API review

Matti Kortelainen

CCE SOP meeting

20 November 2024

General

- CMS found the API review a very useful exercise
 - Allowed us to give early feedback on the API we expect to use for years to come
 - And see the feedback being taken seriously. Thanks!

CMS' points

- `RNTupleView<T, bool>` could be separated into two distinct classes for the two values of `bool`, e.g. `RNTupleView<T>` and `RNTupleUserOwnedView<T>`.
 - Addressed in [root-project/root#16556](#) (also issue in [root-project/root#16321](#))
- For reading `REntry` is created with `RNTupleModel`, but for writing `REntry` is created with `RNTupleWriter`. We would suggest being consistent between the two.
 - Intention is have symmetric reader and writer API wrt. `REntry` handling
- Configurability of `RClusterPool` in similar ways as `TTreeCache`
 - Tracked in [root-project/root/#16325](#)
- Abilities to tune memory usage in `RNTupleWriter`
 - Adaptive page size algorithm was merged. First results on MiniAOD were promising. Further knobs would still be appreciated.

CMS' points (2)

- Inability to use `RNTupleParallelWriter` as long as it is restricted to have only one `Writer` per file
 - Clear guarantees about the locking around `TFile`
 - New method `FillNoCommit()` allowing framework to control the time of `TFile` access
- For every `RNTupleParallelWriter Fill()` call, need to know what entry number that `Fill()` call corresponds to
 - New staged cluster committing should help

- We will look into these improvements in the coming months

CMS' points (3)

- Would like to avoid passing dummy `std::shared_ptr<void>` to `RNTupleReader::GetView<void>()`
 - Was promised to be addressed in the future
- Having to have two separate loops to call `RNTupleModel::AddField()` and `RNTupleModel::GetToken()` feels suboptimal
 - Was promised to be addressed in the future
- `REntry::RFieldToken` not having a default constructor is somewhat inconvenient
 - Was promised to be addressed in the future
- Documentation
 - Relationship between `RNTupleModel`'s `GetToken()` and freezing
 - Projected fields
 - Entry invalidation when `RNTupleModel` is extended