ROOT Workshop I/O Feedback

Xavier Valls for the I/O sub-team

ROOT Workshop 2018
Sarajevo, Bosnia and
Herzegovina

ROOT
Data Analysis Framework
https://root.cern

Bulk I/O & Modularity

- uproot has shown that Bulk I/O can bring performance
  - but the success was also motivated by the programming model and ease of installation
- Should we have an independent library with only (Bulk) I/O?
- Reopen the discussion about user interfaces for Bulk I/O
  - things that were not there before offer new possibilities to converge on something: RTensor (C++), AsMatrix (implicit Bulk I/O),...

TFile Memory Management / Ownership

- People are confused by the fact that TFile can delete objects when closing it
  - Will be solved by ROOT 7

- Python users are even more puzzled because the Python memory management also comes into play
  - Wait for Wim's proposal for a solution

RForest

- Ownership model should be clear
  - When constructing a model, who owns the tree inside the tree?

- RForest should provide two interfaces
  - For users: templated, type-safe
  - For frameworks: with void*, to give more flexibility

Compatibility

- Forward compatibility is rarely an issue, backwards compatibility always is
- Provide meaningful messages when trying to read something too new for the user’s ROOT version

Advice on how to read

- Several options to read data
  - SetBranchAddress + GetEntry
  - TTreeReader
  - Now also RDataFrame
- The ROOT team should give advice/recommendation on what to use and when
  - Do not use the old way in the tutorials
- When searching on Google, new content should be prioritized. Linking to “Latest” may help.
- Fix, remove or reorganize old sections of the ROOT site

Compression algorithms

- Everyone agreed that they should be kept in a rather small subset, not to introduce too much variability

- Only adopt a new one if it brings a substantial advantage

Future I/O

- Should be extensible
- Should have good documentation
- Should have versioning

Metadata, binaries

- From people that do analysis: there should be some way to add metadata in ROOT files
  - How the file was produced and when
  - Done today e.g. with TObjString written in the file

- Users should be able to write binaries in a file, to make it more self-contained
  - Like a zip file

Streamers

- Support for streaming of shared_ptr
  - Included by Philippe in list of priorities for 2018
- Get rid of virtual methods in the streamers
  - Expect 2x improvement in performance

Other feedback

- Type safe templated implementation of TFile::Get()
- In remote access, file chunks are to small (TTreeCache setting not appropriate?)
- Stream operator for TFile
- Add Windows support for DaViX