IRIS-HEP Fellowship Proposal
Pratyush Das

Duration : 24th June 2019 to 16th September 2019

Adding ability to write TTrees to uproot

As an IRIS-HEP undergraduate fellow, I will be working on uproot, a software for reading and writing ROOT files in Python with the help of the Numpy library. Unlike the standard C++ ROOT implementation, uproot is strictly an I/O library, intended to stream data into other third party libraries in Python. Other ROOT file readers in Python like PyROOT and root_numpy rely on the C++ ROOT implementation but uproot does not. Instead, it uses Numpy calls to rapidly cast data blocks in the ROOT file as Numpy arrays.

Until recently, uproot could only read ROOT files. As a DIANA/HEP undergraduate fellow in the summer of 2018, I added functionality in uproot to write strings and histograms to ROOT files. However, one of the most commonly used ROOT classes, TTrees, still cannot be written by uproot. The serialization of TTrees in ROOT is different from the serialization of strings and histograms in that TTree data are not contained within a single object but distributed across the file in independently accessible baskets, making it a new and non-trivial problem to be solved.

I will carry out the work under the mentorship of Jim Pivarski at Fermilab. In the week leading up to the beginning of the project, I will be refamiliarising myself with the uproot codebase and extending the histogram writing ability to include TH2*, TH3* and TProfile histograms. I will also add support for writing TGraph and TMultiGraph, which are serialised differently from histograms but do not have the complexity of TTree's floating baskets, as well as adding support for compression.

The main work on TTrees would involve understanding how C++ ROOT serializes them by examining the bytes from ROOT files, the ROOT C++ code, uproot's reading code, and Go-HEP's reading and writing code and then implementing an independent TTree-writer in Python in uproot. The completion of the project would enable users of uproot to share unbinned datasets with ROOT users and make uproot a more complete ROOT I/O software.

Possible future work in uproot after the completion of my project would include fine-tuning the generation of streamers to only depend on the types of objects being written to the ROOT file and implementing a more efficient block management system for the handling of space allocation when objects with the same name are written to a ROOT file.

Proposed timeline

Week 1

- Finish adding support for TH2*, TH3*, TProfile, TGraph and TMultiGraph.
- Add ability to write compressed objects to a ROOT file.

Week 2-4

- Create a template in uproot for writing TTrees to ROOT files.
- Understand how C++ ROOT serializes TTrees.

Week 5

- Add ability to write a TTree with simple data types to a ROOT file.

Week 6

- Add ability for TTrees to store data in multiple baskets distributed across the file.

Week 7

- Add ability to store TTree data in jagged arrays.

Week 8-11

- Understand ROOT’s splitting algorithm.
- Generalise uproot’s reading algorithm to automatically recognize more split objects.

Week 12

- Clean up codebase.
- Solve any lingering issues.
- Create a jupyter notebook tutorial.
- Prepare a presentation on the work done.