IRIS-HEP Proposal: CMSSW Generating ROOT RNTuple NanoAODs

The CMSSW framework provides the central software components for data processing tasks of the CMS experiment at the LHC [1]. One of the data products it generates are "NanoAOD" files. These are an ultra-compact representation of CMS event data (1-2kB/event). Yet they carry enough information to serve the needs of the majority of physics analysis groups [2].

The ROOT RNTuple project provides a new prototype I/O system for reading and writing HEP event data in a columnar on-disk layout. It has been engineered from scratch as a major part of the ROOTv7 developments and is designed for maximum efficiency for the HEP data analysis use case [3]. RNTuple has two main influences. It is guided by the ROOT TTree format and its 20 years of production experience but also draws from recent advances in industrial Big Data file formats (e.g. Parquet, Arrow). The RNTuple prototype has shown significant improvements compared to the current production representation of nanoAOD files. First measurements suggest 10-20% smaller files after LZMA compression and over 4 times faster read throughput for SSD devices.

The goal and main deliverable of this project is the development of a CMSSW output module that uses the ROOT RNTuple interface to generate nanoAOD files in the RNTuple format. A new RNTuple output module can validate the RNTuple class design in the context of a large, production-grade experiment framework. It would also allow for large-scale measurements comparing the TTree and RNTuple formats for compact AODs. In particular, we expect the following results:

1. Validation, and where necessary, adjustments, of the RNTuple C++ interfaces in the context of a large, multi-threaded HEP experiment framework,
2. Validation of the RNTuple format and event data model,
3. A statistically sound comparison of RNTuple and TTree storage consumption (among other important metrics) facilitated by the production of semantically equivalent data sets in both formats,
4. A first evaluation of parallel RNTuple writing (1 event cluster per thread).

This project is a natural continuation of my Google Summer of Code (GSoC) project, which focused on fast merging of RNTuple files and asynchronous file I/O. As part of the GSoC project I implemented essential building blocks for a potential CMSSW output module, including RNTuple metadata handling and low-level I/O routines that can be reused for parallel writes. The CMSSW framework is a prime candidate for early RNTuple integration due to its strong focus on multi-threaded workflows and the fact that the ROOTv7 classes, including RNTuple, are already part of the CMSSW software stack. The project would be co-supervised by Jakob Blomer (CERN) for the RNTuple aspects and Dan Riley (Cornell) for the CMSSW aspects.

Timeline

This project is scoped for 6 months of half-time work and takes into account my familiarity with the RNTuple classes. The project will deliver the following items with a reserve period of 3 weeks:

1. Creation of a new CMSSW output module skeleton that takes event data as an input and produces an empty ROOT file as an output (~4 weeks)
2. Implementation of the nanoAOD event data model using RNTuple classes resulting in the RNTuple output of the ~1500 data members of the "Events" collection (~4 weeks)
3. Implementation of the nanoAOD auxiliary data output in the RNTuple format (~3 weeks)
4. Implementation of parallel RNTuple writing based on event clusters (~6 weeks)
5. Documentation, testing, and first measurements (~4 weeks)

The proposed timeline for this project is November 2020 to April 2021. During this period, I will be enrolled in one course per semester (half of a full course load). I will also be conducting research activities for my Master's thesis.

References

[1] *CMS Offline Software.* https://github.com/cms-sw/cmssw

[2] *A further reduction in CMS event data for analysis: the NANOAOD format.* https://doi.org/10.1051/epjconf/201921406021

[3] *Evolution of the ROOT Tree I/O.* https://arxiv.org/abs/2003.07669