IRIS-HEP Project Proposal: Enable Dask interopability with xrootd-accessible storage systems

Student: Scott Demarest

Mentors: Jim Pivarski, Nick Smith

Project duration: 1 June to 10 August 2022

Proposal

Dask is one of the most popular parallel computing Python libraries used in HEP data analysis. Among other things, it allows easy access to data storage systems both local and remote. For remote storage services (like Amazon S3 or Google Drive) a Dask user need only input the URL prefixed with corresponding protocol code and credentials. This way, the user can spend more time doing physics and less time learning the ins and outs of each file system. Dask can easily scale up to the multi-terabyte datasets typical in HEP making it a powerful and convenient tool.

All this is enabled by fsspec, a Python library that provides a common interface for many data storage systems. By default, it comes with implementations for many common storage systems (such as Libarchive or Arrow's HadoopFileSystem) and more can be installed separately. As of now however, there is no implementation for xrootd accessible storage systems. There do exist Python bindings (pyxrootd) but the API differs from that required by fsspec and Dask. There are differences in function names as well as functions not shared between the two APIs. A proper fsspec implementation is required for Dask to easily use xrootd systems. Since xrootd is commonly used in HEP, an xrootd implementation of fsspec would be a convenient and valuable addition to the scientific Python ecosystem.

This project proposes to develop a fsspec-xrootd middleware software package, and release it as part of either Coffea or Scikit-HEP. Over the course of ten weeks, the package, documentation, and a presentation will be produced. Demonstrations will also be made which utilize the package and the rest of the ecosystem to perform a data analysis task. This will be an opportunity for me to do some particle physics with the tools I helped to make. It will also be an educational experience in which I develop skills in Python development, packaging, and open-source software maintenance. By the end of this project, I will have familiarized myself with the tools and services used by high energy physicists.

Timeline

W1: Familiarize with Dask, fsspec, and xrootd.
W2: Set up package on github, plan out package structure, look at other fsspec implementations for reference.

W3-6: Implement and test synchronous functions from the fsspec AbstractFileSystem base class that are in common with pyxrootd.
W7-9: Implement asynchronous functions and perform benchmarks to understand the performance differences between the two. Start to build a physics analysis demo for the library.
W9-10: Finish documentation, demos, and prepare presentation.