Enabling Dask interopability with xrootd-accessible storage systems

Lightning Talk by Scott Demarest (Florida Institute of Technology)

Mentors: Nick Smith (FNAL) and Jim Pivarski (Princeton)

Dask: Popular parallel computing Python library

- Big data handling
- Dynamic task scheduling

XRootD: Data storage framework common in HEP

ROOT File: Saves and loads complex data structures quickly

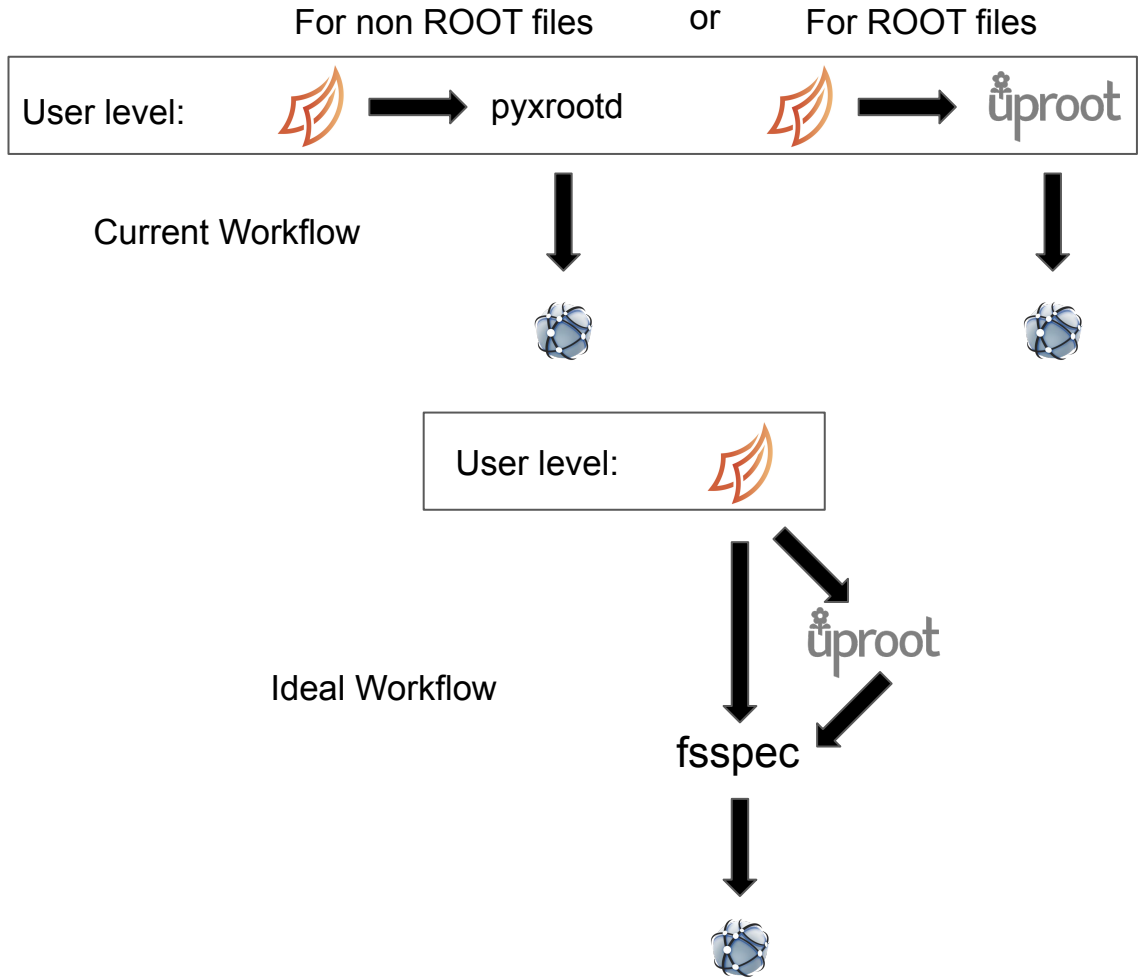
The Problems

1. Dask can't directly use XRootD accessible storage systems
2. Dask can't directly open ROOT files

Current workarounds require Dask users to use and implement special libraries. This takes time away from doing physics.

Ideally, Dask users should only have to use Dask and everything else should be handled in the background.

With the right interfaces, existing tools can be arranged in layers such that each layer solves a specific problem.



The Solutions

A. Already exists. Dask uses the fsspec standard API to access many file systems.

B. Does not exist. Need an fsspec-xrootd implementation. To be released as a package.

C. Proof of concept by Doug Davis exists. See [ContinuumIO/dask-awkward#44]

D. Does not exist. Need to add fsspec.py to Uproot source. Released as PR to Uproot

The result of this project will be the creation of interfaces B and D. This will allow Dask to handle XRootD and ROOT files in the background, making Dask more user-friendly.

fsspec allows for read ahead caching and asynchronous functionality. These features may result in improved performance.