

# Enabling Dask interoperability with xrootd-accessible storage systems

Fellows Talk by Scott Demarest (Florida Institute of Technology)

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



# What's What



Dask: Popular parallel computing Python library

- Big data handling
- Computer Cluster Scheduling

fsspec

fsspec: A common interface for all file systems

- Used by Dask, pandas, and other packages



XRootD: Data storage framework common in HEP

- CERN's Open Data for example

# The Problem

```
import dask.dataframe as dd
```

```
df = dd.read_csv('URL')
```

```
df.head()
```

URL can refer to any of the remote file systems supported by fsspec like google drive, jlab and more.

XRootD urls are not supported by fsspec.

Users need external packages and extra code.

# The Solution

We created a package: fsspec-xrootd

Published on PyPI and ready to use.

Once installed, Dask can use XRootD urls.

**fsspec-xrootd 0.2.1**

```
pip install fsspec-xrootd
```



# An Example

```
import dask.dataframe as dd
```

```
df = dd.read_csv('root://...testfile.csv')
```

```
df.head()
```

Dask accepts XRootD urls now.

No extra user code needed.

Note: XRootD urls start with “root://”

# What It's Doing

While the project goal was to make Dask work with XRootD, our package has nothing to do with Dask in particular.

fsspec-xrootd is the interface between fsspec and the XRootD server.

This means that you can directly use fsspec to access your XRootD data.



fsspec



fsspec-xrootd



# Using fsspec and fsspec-xrootd

```
with fsspec.open("url", "rb") as f:  
    f.read()  
    f.seek(40)
```

Our package supports the full range of fsspec functionality

File and file system level operations

```
fs.ls("PATH", False)  
fs.info("PATH")
```

Can run concurrently with asyncio

```
r = await fs._cat_ranges(paths, starts, ends, batch_size=100)
```

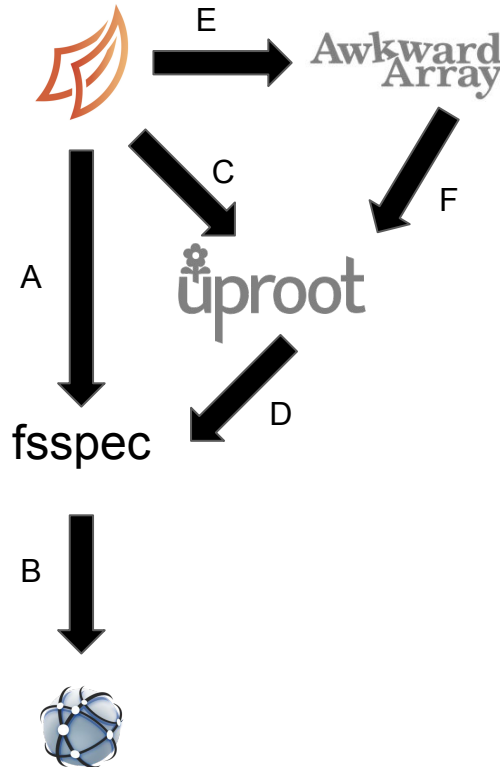
# Completing the Ecosystem

This package is one part of a bigger picture.

The end goal is to link Dask, Uproot, fsspec, and XRootD into one, seamless experience.

This way, Dask will be able to both access XRootD data and interpret .root files.

dask-uproot (C): Kush Kothari  
fsspec-uproot (D): Nick Smith  
dask-awkward (E): Doug Davis





# My Experience

I was taken through the process of developing a software product

- Edge cases
- Fault tolerant
- Unit tests
- Continuous Integration

I sharpened my Python abilities

- Concurrent programming
- OOP
- Reading Documentation
- Using GitHub

# In Sum

Dask/fsspec users can directly access XRootD data

fsspec-xrootd is working and is published on PyPI

Full Dask ecosystem coming soon

Just pip install fsspec-xrootd

Any Questions?