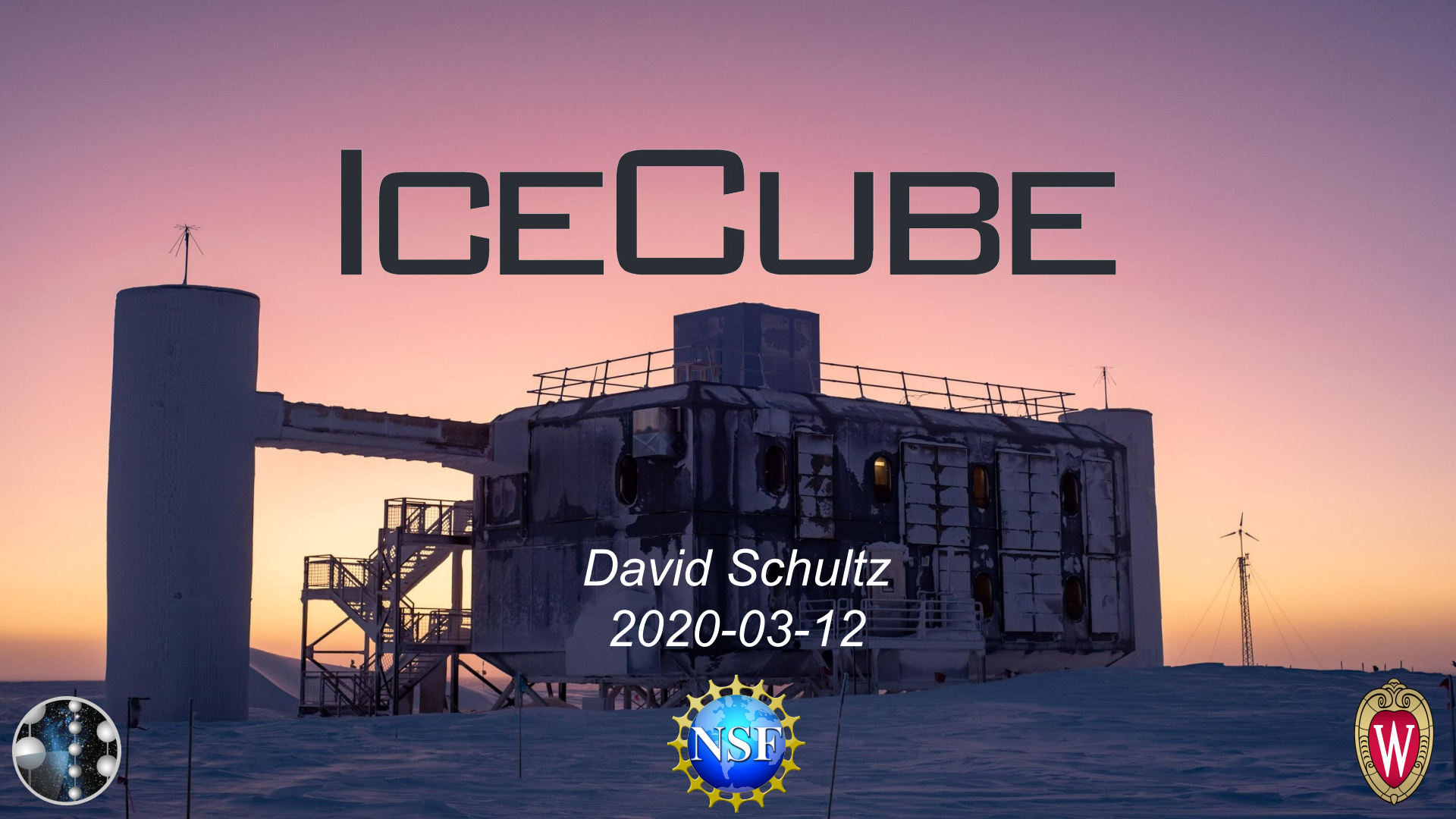
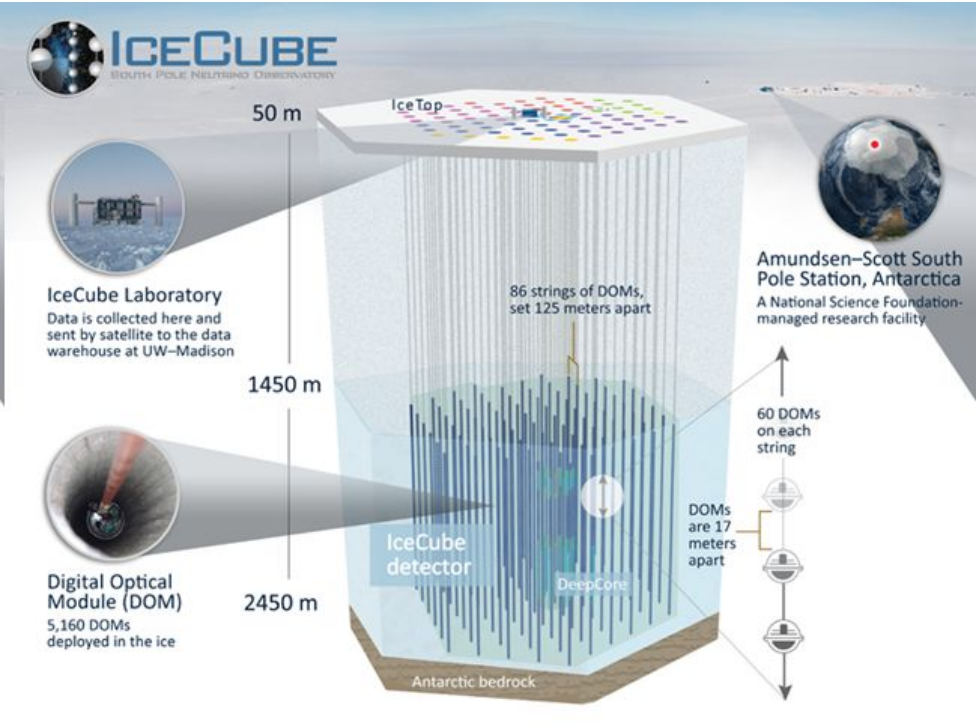
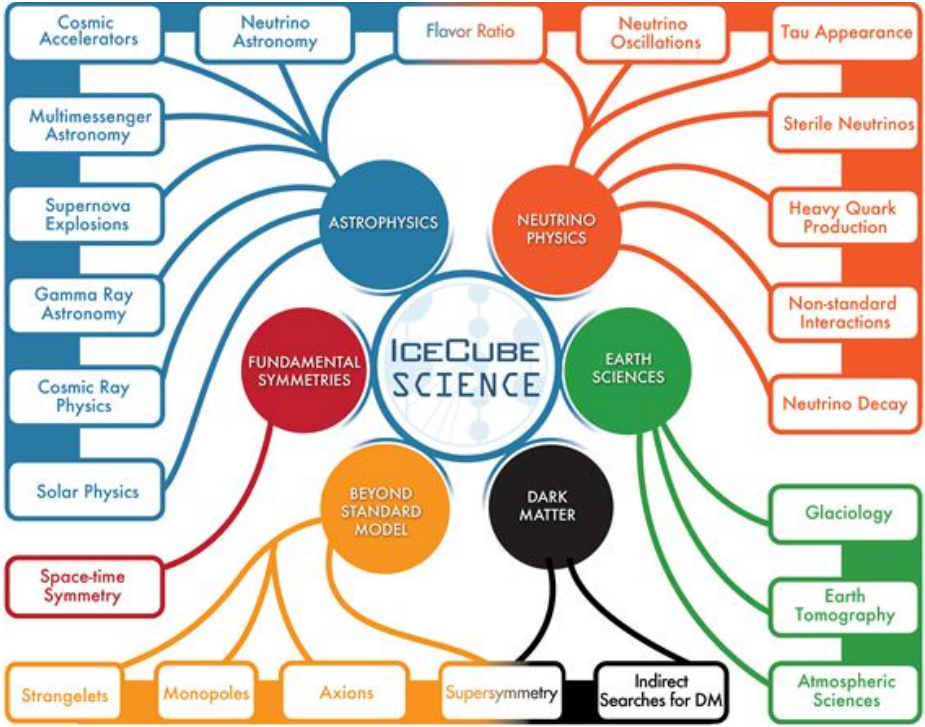

IceCube
David Schultz
2020-03-12
NSF

Outline

- What is IceCube
- Areas we want to use Rucio
  - Some history
  - Where we are now
- Pain points

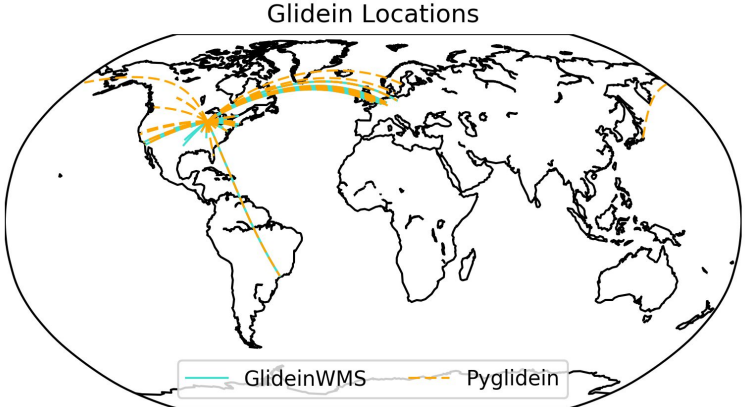
IceCube Detector and Science

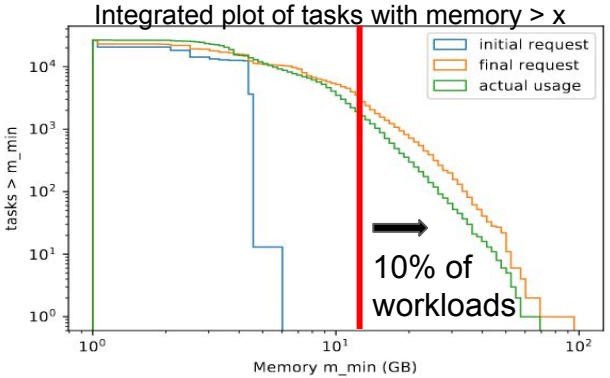




IceCube Computing

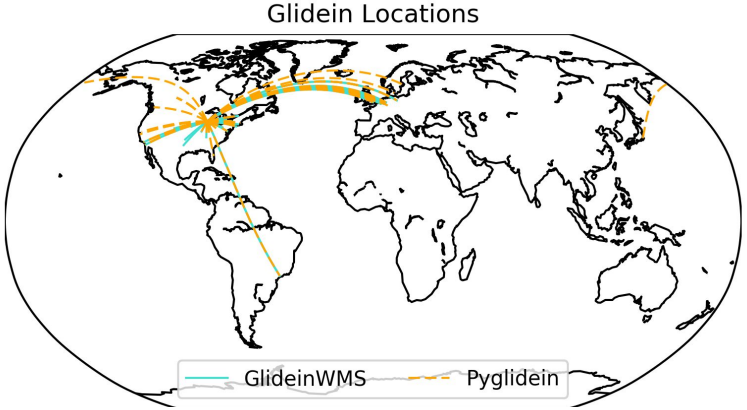
- Global heterogeneous resources pool
- Mostly shared and opportunistic resources
- Atypical resources requirements and software stack
  - Accelerators (GPUs)
  - Broad physics reach - Lots of physics to simulate
  - Data flow includes leg across satellite
  - “Analysis” software is produced in-house
    - “Standard” packages, e.g. GEANT4, don’t support everything or don’t exist
    - Niche dependencies, e.g. CORSIKA (air showers)

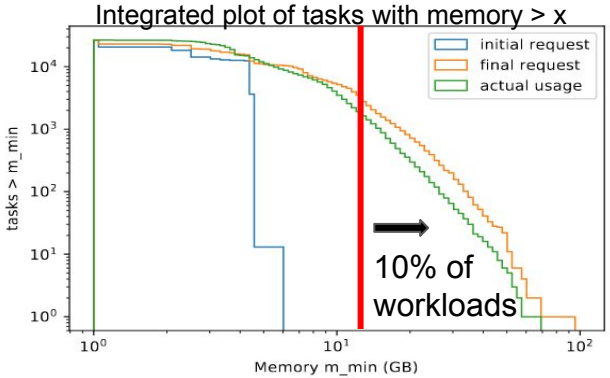




IceCube Computing

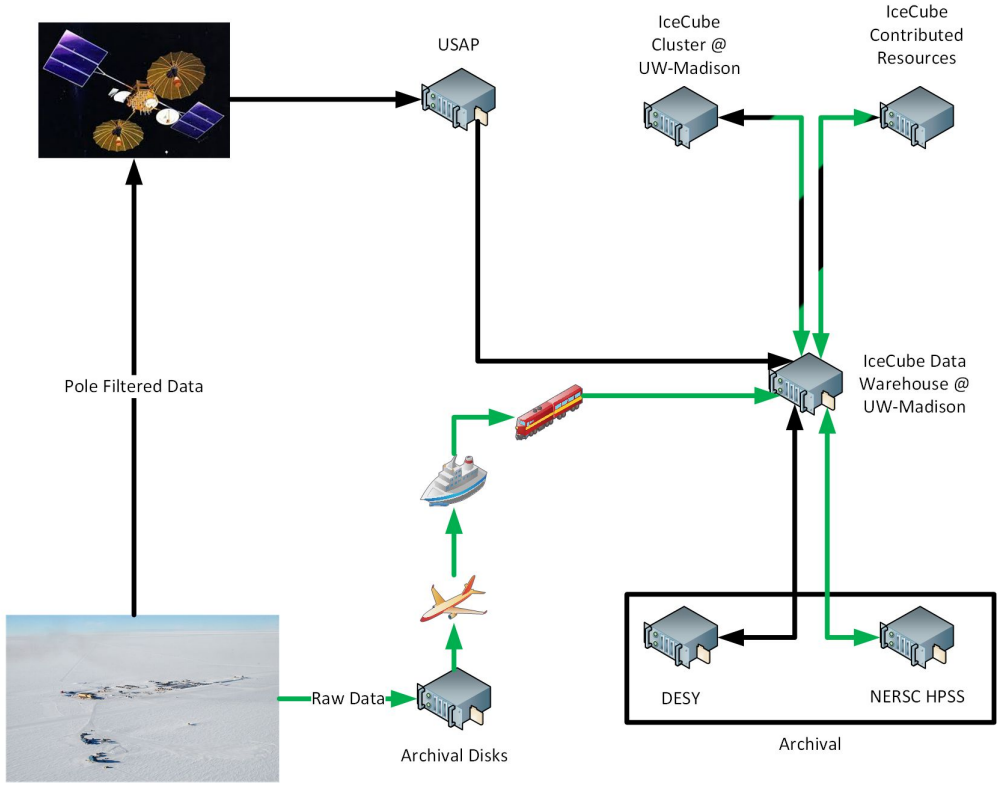
- Detector up time at 99+% level
  - There are no downtimes
- Built in natural medium
  - Interesting calibration problem
- Significant changes of requirements over the course of experiment
  - Accelerators
  - Multimessenger Astrophysics
  - Alerting
  - ….





Data Flow and Processing

- Pole Filtered Data arrives via satellite - Arrives at UW-Madison and is reduced further to higher levels
- Raw data is written to archival disk at pole, retrieved once a year
- Raw data is archived at National Energy Research Scientific Computing Center (NERSC)
- Filtered data is archived at Deutsches Elektronen-Synchrotron (DESY)



Data Management

- All active data stored in one place
  - UW-Madison currently stores 7.5 PB on online storage
    - 2.1 PB (72M files) of detector and common reconstructed data
    - 3.1 PB (91M files) of simulation data
    - 2.3 PB (570M files) of higher level analyzer data
- Currently using basic Linux tools to manage storage
  - <1 person needed to support this, common tools
  - Rucio seems over-engineered for single-site management

Where we'd like to use Rucio

- Archiving is the major use case
  - Raw detector data to NERSC (tape) - 350 TB/year
  - Filtered detector data to DESY (tape) - 30 TB/year
  - Higher level reconstructed data to DESY (online) - 120 TB/year

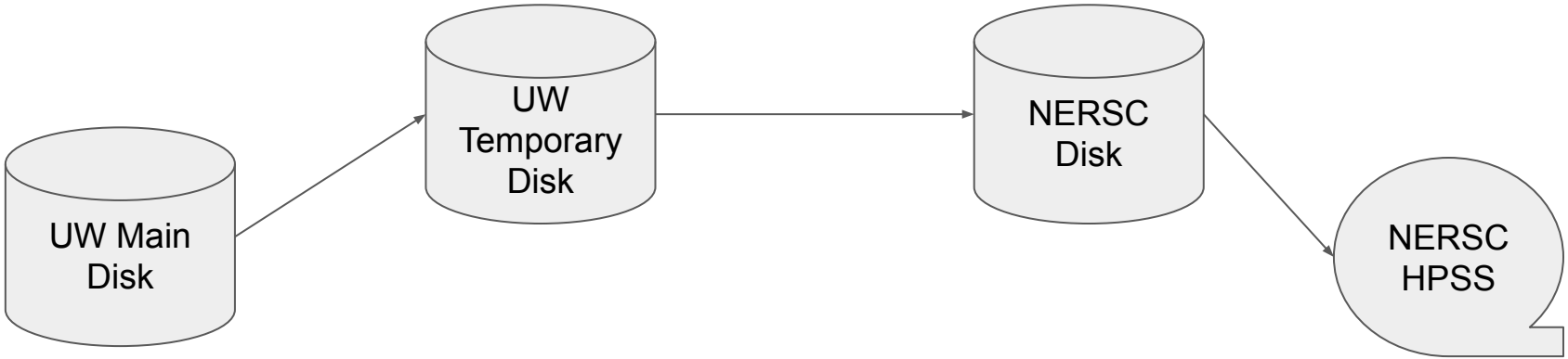
Where we'd like to use Rucio

- Archiving is the major use case
  - Raw dete (tape) - 350 TB/year
  - Filtered d (tape) - 30 TB/year
  - Higher level reconstructed data to DESY (online) - 120 TB/year

Must be bundled into ~500 GB files for the tape systems

Where we'd like to use Rucio

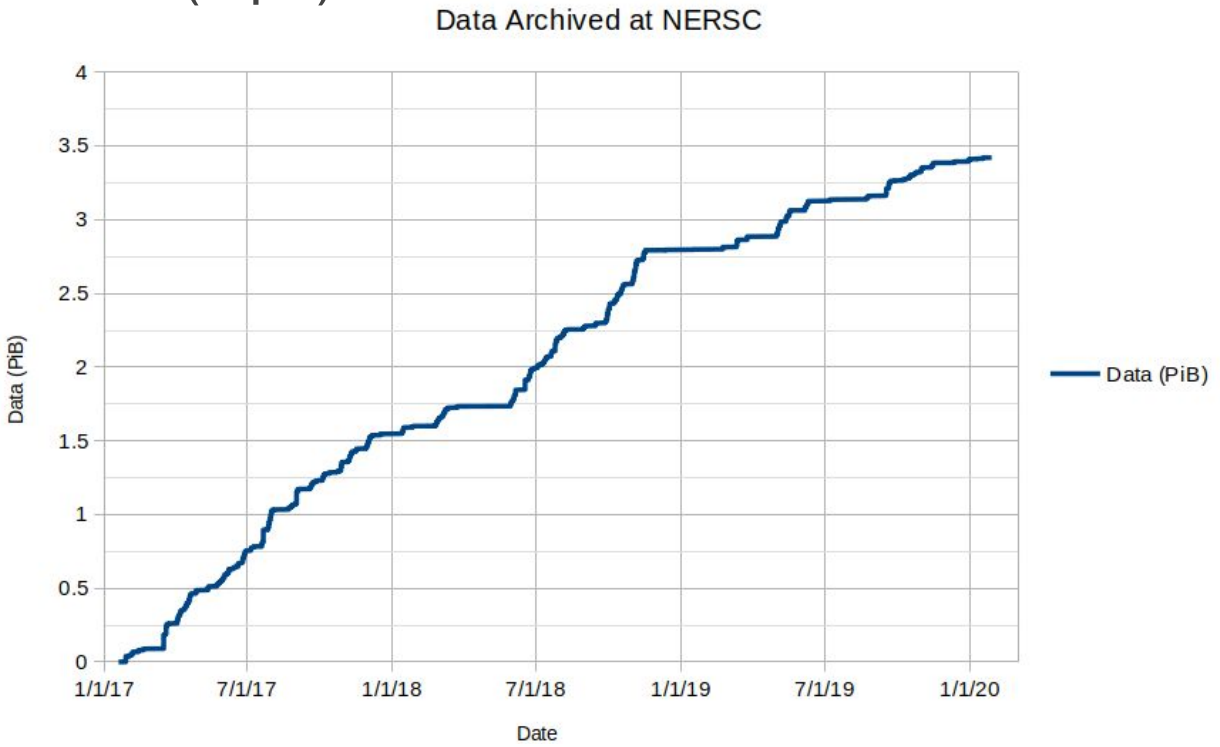
Raw detector data to NERSC (tape)



Where we'd like to use Rucio

Raw detector data to NERSC (tape)

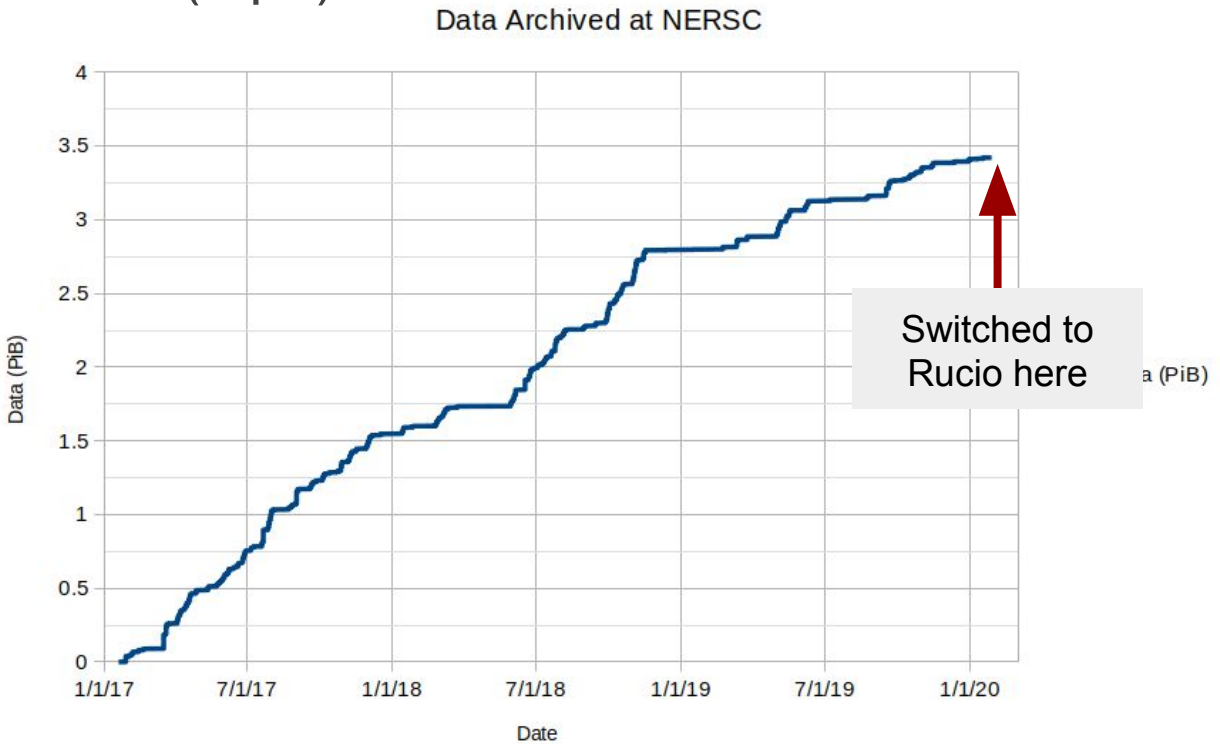
We have some history at NERSC, using fragile scripts and globus online



Where we are using Rucio

Raw detector data to NERSC (tape)

A properly designed system is replacing these scripts, with the transfer step switching to Rucio in the last weeks



Where we are using Rucio

Filtered detector data to DESY (tape)

Scheduled to begin in the next months, now that NERSC is understood.

Where we are using Rucio

Higher level reconstructed data to DESY (online)

Transitioned to Rucio in mid-2019

So far, have moved 360 TB of past years' data, and actively sync current data daily

- This is the only workload that only uses Rucio, as it is just replicating a directory

Pain Points

Learning about Rucio

- Documentation coverage is mixed
  - “Some of the concepts and terminology are explained well to newcomers, others are still quite opaque until one gets more familiar with the internals of Rucio.”
  - Technical details sometimes missing - need to go to source code
  - Operational issues could use more coverage:
    - What to do with a stuck file?
    - How do daemons work together?

Pain Points

Attempting to use Rucio

- Some surprises when using the CLI
  - Example: `rucio list-dids <scope>:*`
    - Does not actually list all DIDs
    - Must add `--filter type=all` to list file DIDs
- Non-orthogonal REST APIs
  - A DID can refer to files, datasets, or containers
  - Creating files cannot be done through `POST /dids`
  - Must use `POST /replicas`
  - Unintuitive for beginners, not well documented

Pain Points

Attempting to use Rucio

- Docker container issues
  - Incomplete configuration files
  - Hard-coded values, like /etc/grid-security
- Ghost objects
  - While you can “delete” an RSE, it still exists in the DB
  - No commands to really delete, or restore it
  - Can’t query for “deleted” RSEs
  - Can’t recreate RSE that previously existed and is “deleted”

Pain Points

Technical difficulties

- FTS transfers
  - Several unexplained FTS issues, never identified
  - Solution was “restart FTS”
- Memory usage
  - While replicating a dataset, observed Rucio using >70GB of RAM to create a replication rule for 2.5M files
  - An instance with 512 GB of RAM was discussed in Oslo
  - These are non-trivial hardware requirements for us

Future

Given the difficulties, three paths:

1. Devote at least one person to Rucio management
   - This isn't likely, as we don't have any extra people
2. Outsource anything involving Rucio somehow?
3. Limit Rucio's role in IceCube (to contain issues)

Conclusions

- IceCube is using Rucio to archive files to two sites
- Several difficulties encountered and overcome
- Will see what the future holds