GridPP
UK Computing for Particle Physics

Rucio and DUNE Data Management at Edinburgh

Wenlong Yuan

GridPP45, 20th Oct. 2020

THE UNIVERSITY of EDINBURGH

Outline

- Rucio activities in DUNE Data Management
- Rucio Monitoring at Edinburgh
- Rucio with Object Storage Systems and Policy Packages

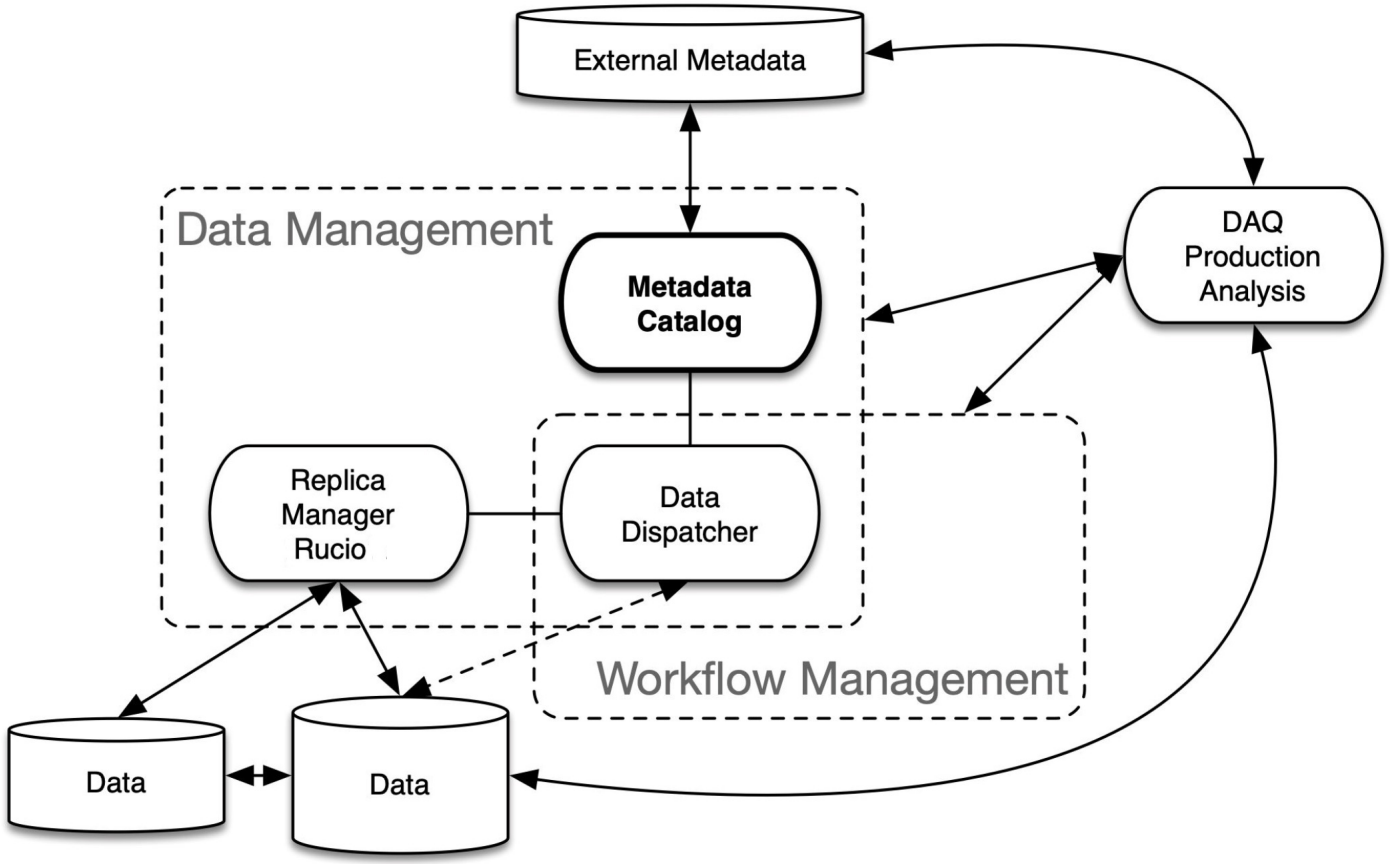
DUNE Rucio and Data Management Currently

- DUNE has been running Rucio since Fall of 2018
- Two ProtoDUNE detectors at CERN, each 5% of full detector size

- 36 compute sites around the world, 13 disk only sites, 4 disk+tape sites
- 17 commissioned Rucio Storage Elements (RSE)
- 13 PB under Rucio management, 1,398,000 DIDs, 3,112,187 replicas
- DUNE detectors output 30+ PB per year

- Ingest of ProtoDUNE raw data still done with legacy system - Serial Access via Metadata (SAM), tell us what it is, and where it is (But not for much longer)

DUNE Data Management Architecture

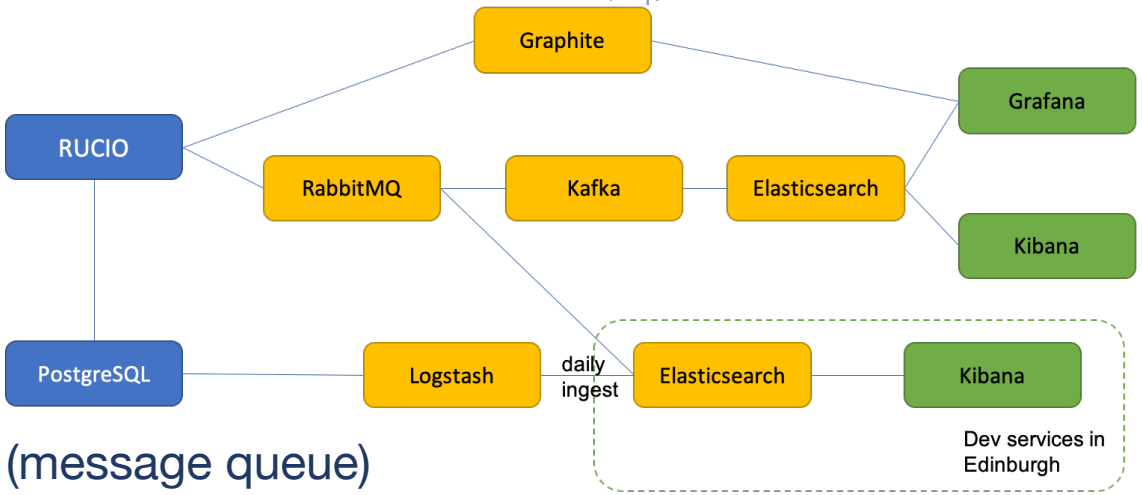


- Using Rucio for data movement
  - Need to change data delivery system so use the Rucio file location info and use Rucio to deliver the file location
- replace 3 main functions of monolithic legacy system (SAM)
  - **Replica manager**
    -> **Rucio**
  - **File Provenance** (Metadata)
    -> Metadata Catalog (**MetaCat**)
  - **Data Delivery / project tracking**
    -> **Data Dispatcher**

DUNE Rucio activities

- DUNE Data Management Ops
  - Rucio clients to move data from point A to point B. (asking for help if things get stuck)
  - Creation and declaration of new Rucio replicas
  - Onboarding new remote Rucio storage elements
  - Interaction with remote sites for transfer
- Lightweight client for REST API
  - Stock client has lots of dependencies
- Getting job input data from Rucio
  - Investigating how to let people get job input data from UK RSEs
  - Getting user and production jobs to find data from closest storage
  - Getting user and production jobs to write to local storage

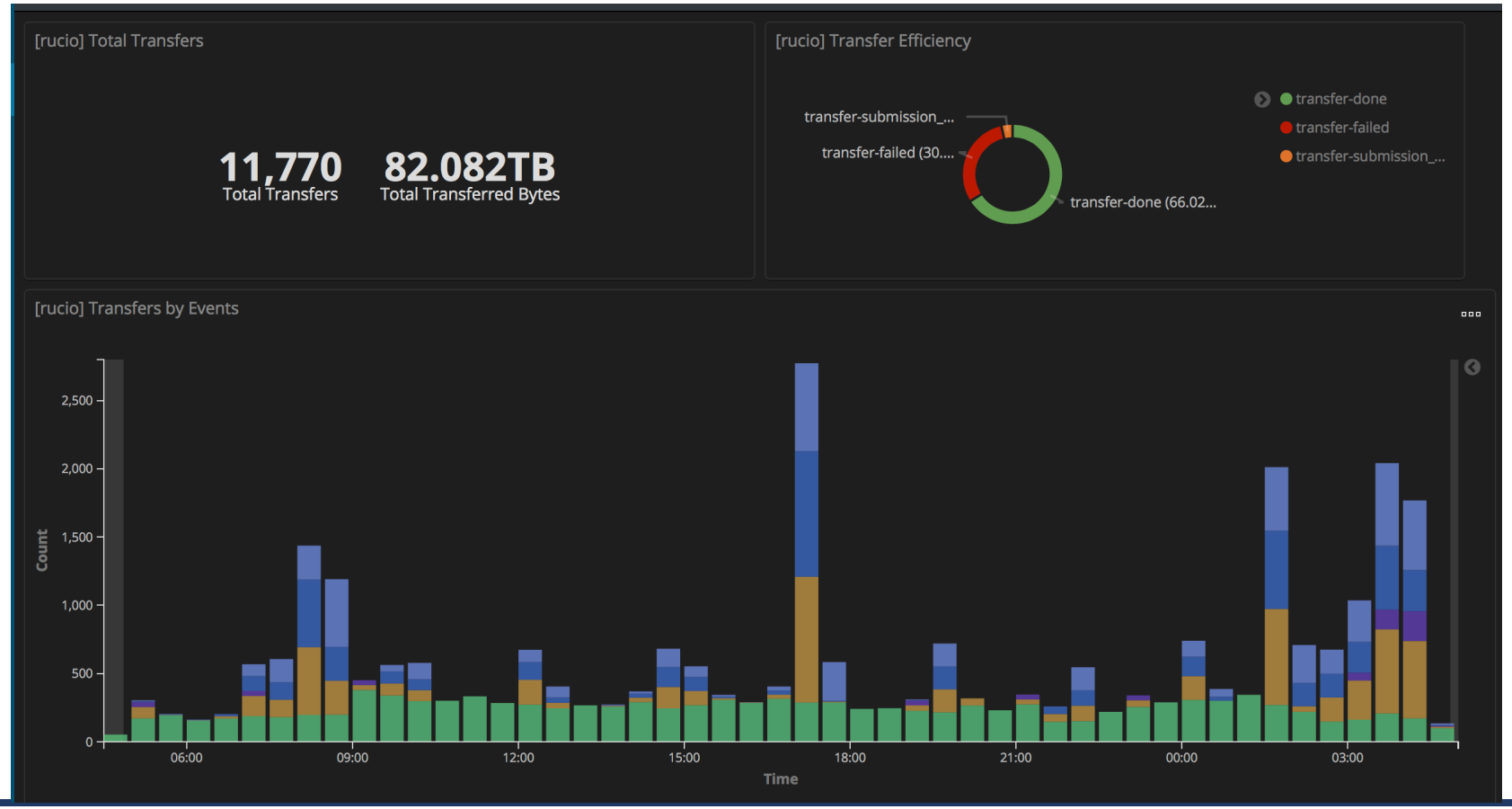
Rucio Monitoring



- Internal metrics – Graphite
- Data transferring / deletion events
  - Rucio daemons generate messages when submitting / staging / queueing / finishing
  - Messages are sent to be cached in the broker (message queue)
  - Messages can be dumped to Elasticsearch and be visualized
- Replica / accounting / client trace
  - The replica location, accounting and client trace data are recorded in the RUICO internal database
  - To efficiently visualize them, DB tables can be dumped to Elasticsearch periodically
    - Use logstash pipelines with jdbc
    - Perform joint queries to resolve RUCIO internal IDs
    - Setup daily dump

Monitoring: Data transferring / deletion

- Rucio Kibana Monitoring. Shows queued, failed, submitted, done.



Monitoring: Replica / accounting / client trace

GridPP
UK Computing for Particle Physics

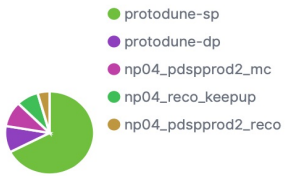
[rucio] Mock _Total dids

1,398,000 DIDs
5.9PB Total bytes

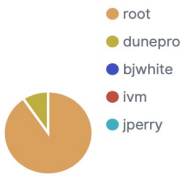
[rucio] total replicas

3,112,187 Total replicas
13PB Total bytes

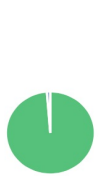
[rucio] DIDs per scope



[rucio] DIDs per account



[rucio] DIDs per did type



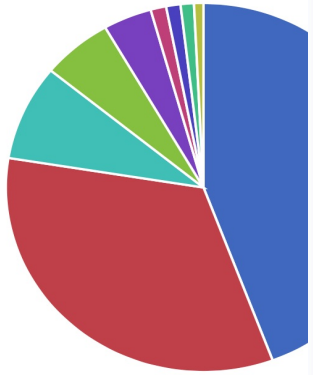
[rucio] DIDs per availability

[rucio] Replicas per site

RSE	Replicas	Total bytes
FNAL_DCACHE	1,377,955	5.9PB
CERN_PDUNE_CASTOR	1,034,222	4.4PB
CERN_PDUNE_EOS	260,421	816.1TB
RAL_ECHO	179,633	844.9TB
MANCHESTER	123,689	463.9TB
PRAGUE	39,558	272.7TB
DUNE_US_BNL_SDCC	36,599	71.8TB
IMPERIAL	33,894	243TB
LANCASTER	24,045	122.3TB

Export: Raw Formatted

[rucio] Replicas pie



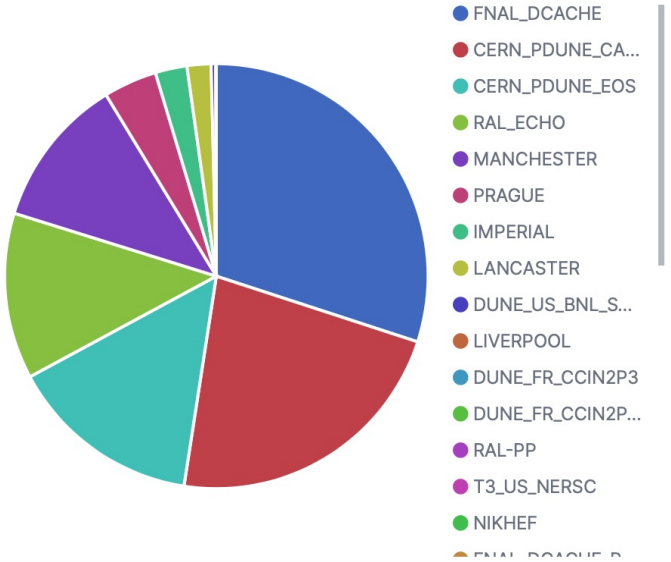
- Monitoring Storage Allocation
  - Info from SRR json (Storage Resource Reporting)
  - Available on DPM sites
  - Need further config on STORM and dCache sites

DUNE RSE USAGE

RSE	Total Allocation	Used	Free	Free(%)
MANCHESTER	980TB	438.2TB	541.8TB	55.287%
RAL_ECHO	909.5TB	844.9TB	64.6TB	7.099%
LANCASTER	150TB	122.3TB	27.7TB	18.467%
RAL-PP	9.1TB	10.1GB	9.1TB	99.892%
LIVERPOOL	1TB	549GB	475GB	46.388%

Export: Raw Formatted

Rucio used space per site pie



Object Storage Systems and Policy Packages
(James Perry)

- URL signature support was added to Rucio core
  - Google Cloud Platform, Amazon S3 and Openstack Swift all supported
  - Code cleaned up and tested for scalability
- Experiment-specific “policy packages”
  - Previously, experiment-specific customizations stored in Rucio core code, not sustainable as more and more experiments use Rucio
    - e.g. permissions model, schema, lfn2pfn and surl functions
  - Move this code into a separate per-experiment “policy package”
  - Python package maintained by experiment, simple to create, documentation available
    - can be installed locally or from a repository
  - Basic support now in Rucio - Introduced in 1.22
  - Multi VO support in progress
    - Each VO can have its own policy package
    - First multi VO policy package PR already merged

Conclusions

- Rucio activities and experiences at Edinburgh
  - Data management ops
  - Monitoring
  - Object storage systems and policy packages core codes dev
- Plan to contribution more to Rucio
  - More activities in data management
  - Getting job input data from Rucio
  - Lightweight client for REST API
  - Support more experiments trying Rucio