Rucio at RAL

Alastair Dewhurst

Abtract
Rucio is the name of the ATLAS distributed data management system. Rucio has been in production since 2013 and has proven to be extremely scalable. It currently manages around a billion files using over 350PB of storage across 120 sites. One of the key features of Rucio is its policy driven approach to data management. This has gained significant interest from other communities.

In April 2018 the Tier-1 setup a Rucio instance on the SCD Cloud for evaluation by SKA. More recently, the Tier-1 has received funding from IRIS to build a production quality Rucio instance and develop the necessary tools to allow multiple experiments to use it.

In this talk I will describe what Rucio does from both the users and admins perspective, the setup at RAL and our future plans.

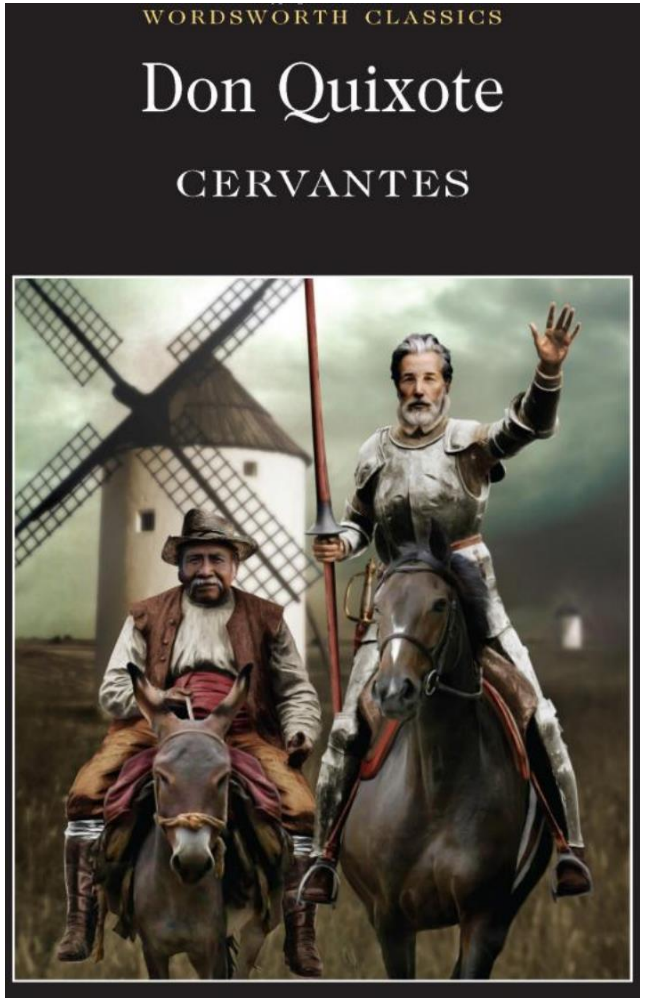
A long time ago…

Don Quixote is the story of a deranged gentleman who believes he is a knight. He famously attacks windmills believing them to be giants.

In the early days of the Grid, operating the ATLAS data management system was described as like: “Don Quixote fighting windmills”.

DQ2 was the data management system for LHC Run 1.

Rucio is the name of the donkey and is the successor to DQ2.

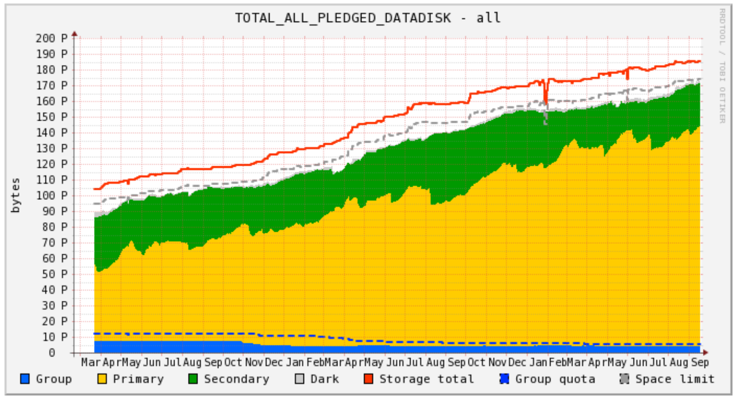


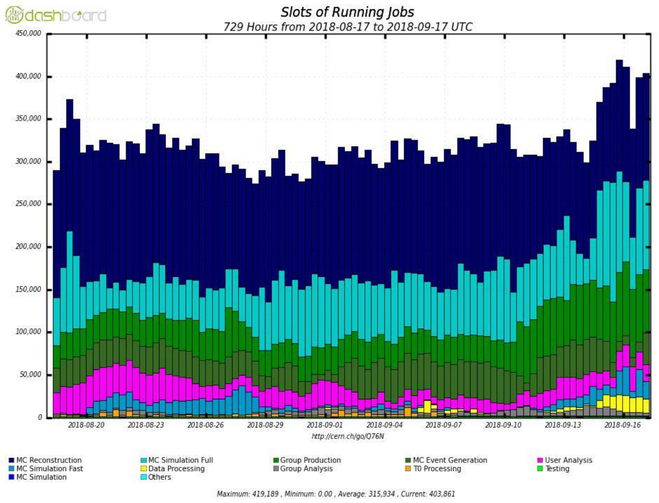
What is Rucio?

- Rucio provides a complete and generic scientific data management service
  - Designed with more than 10 years of operational experience in large-scale data management!
- Rucio manages multi-location data in a heterogeneous distributed environment
  - Creation, location, transfer, and deletion of replicas of data
  - Orchestration according to both low-level and high-level driven data management policies (usage policies, access control, and data lifetime)
  - Interfaces with workflow management systems
  - Supports a rich set of advanced features, use cases, and requirements
  - Large-scale and repetitive operational tasks can be automated

Rucio for ATLAS

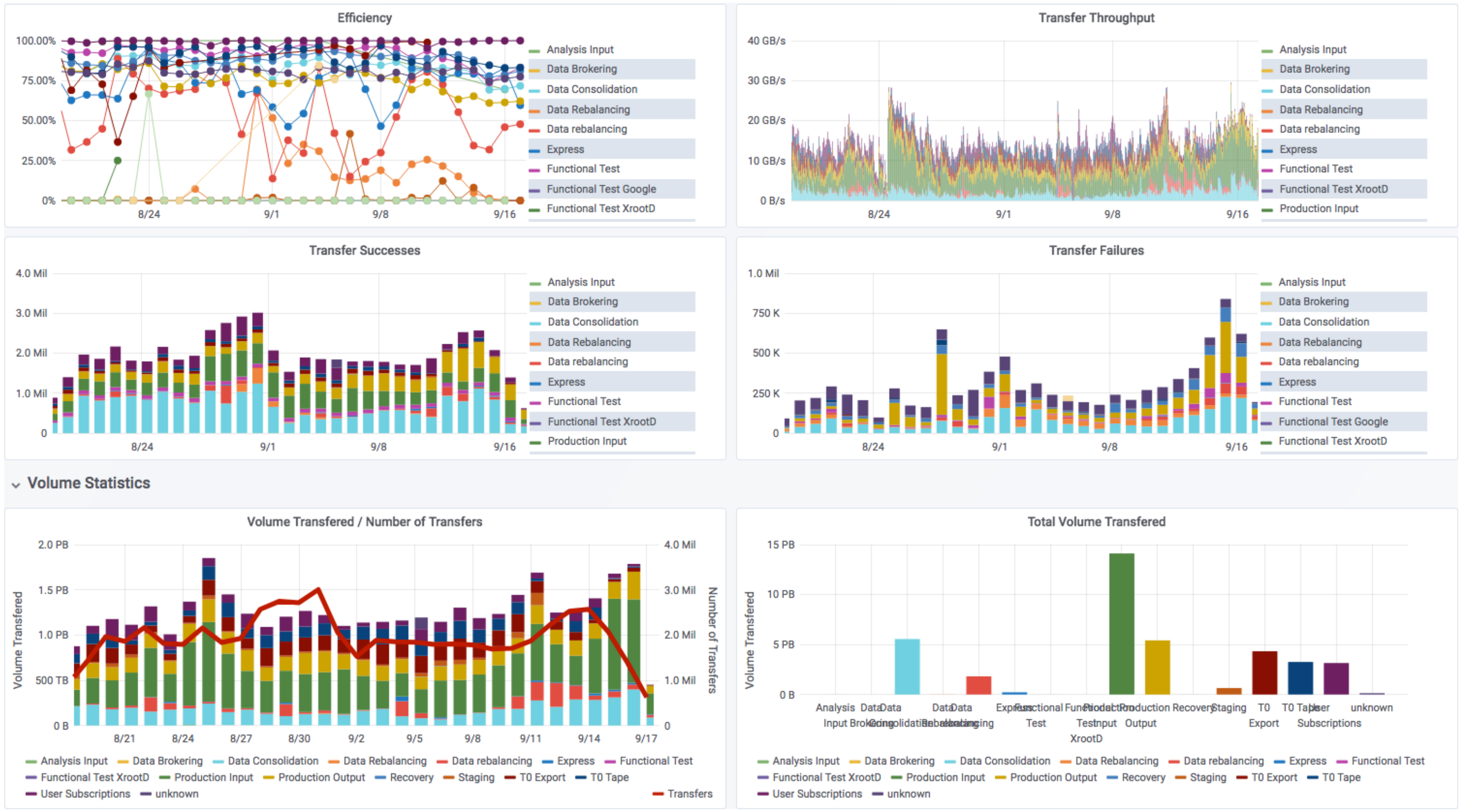
- Rucio is the Distributed Data Management service used by ATLAS.
  - Phased in to production from 2013 - 2015.
- Manages 350PB+ of data across 130 sites.





ATLAS Transfers

ATLAS Transfers via FTS in the last 30 days



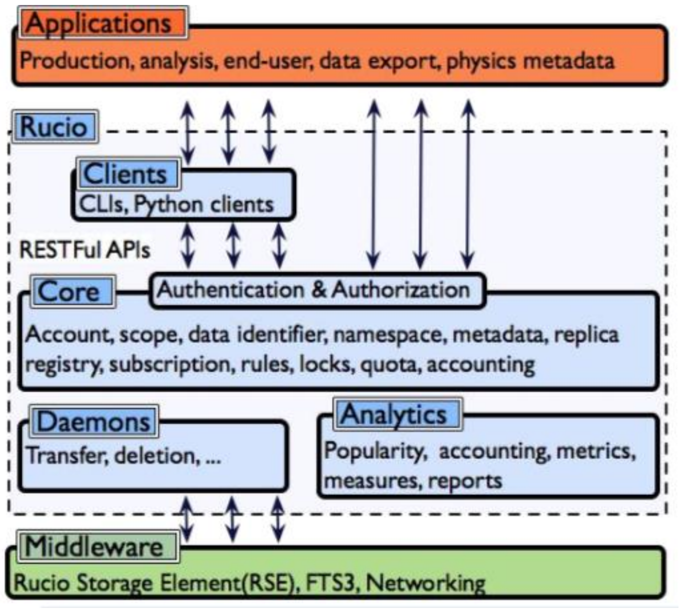
Other Rucio users?

- The Xenon1T and AMS experiments use Rucio.
- ASCG run a Rucio instance for local users.
- CMS are looking at switching from PhEDEx to Rucio for their data management for Run 3
    - LIGO, SKA, DUNE and others are looking at it too.
- Why the sudden interest in Rucio?
    - Proven to work.
    - Rucio developers can’t rely exclusively on ATLAS for funding.
    - WLCG community is looking at future challenges.

Using Rucio

Architecture

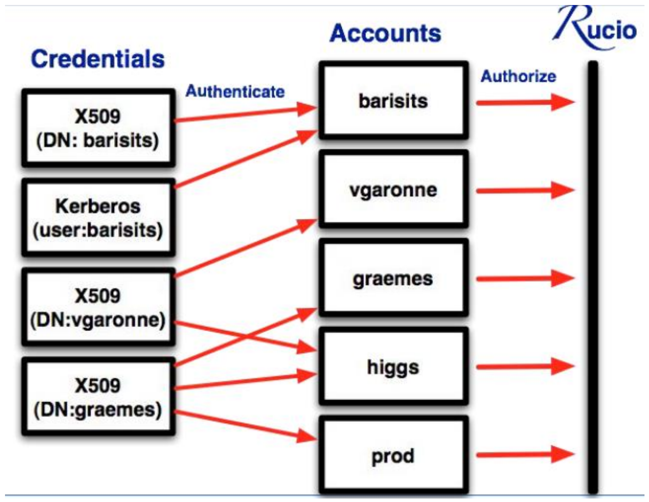
Fully built on open standards and frameworks! To follow the advances with a flexible design with no dependence on particular implementation.

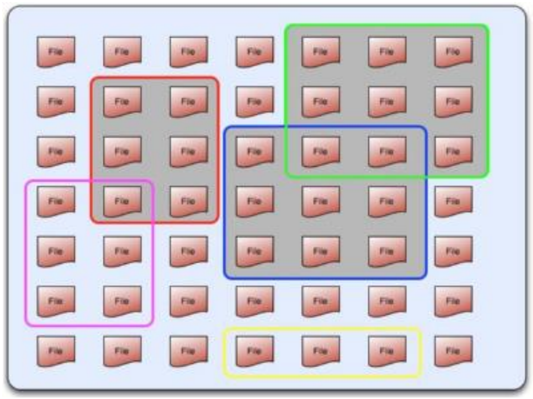

Applications
Production, analysis, end-user, data export, physics metadata
Rucio
Clients
CLIs, Python clients
RESTFul APIs
Core
Authentication & Authorization
Account, scope, data identifier, namespace, metadata, replica registry, subscription, rules, locks, quota, accounting
Daemons
Transfer, deletion, ...
Analytics
Popularity, accounting, metrics, measures, reports
Middleware
Rucio Storage Element(RSE), FTS3, Networking

- Servers
  - HTTP REST/JSON APIs
  - Token-based authentication
  - Horizontally scalable
- Daemons
  - Orchestrates the collaborative work e.g., transfers, deletion, recovery, policy
  - Horizontally scalable
- Persistence
  - Object relational mapping
  - Oracle, PostgreSQL, MySQL/MariaDB, SQLite
- Middleware
  - Connects to well-established products,E.g., dCache, EOS, S3, ..

Namespace

- Rucio account is the entry point for assigning privileges.
  - The same user credential can be used to connect to several accounts
- Files are grouped into datasets
- Datasets are grouped into containers
  - Datasets only hold files
- Containers are grouped into containers
  - Containers only hold datasets or containers
- Collections can be organised freely
  - Files can be in multiple datasets
  - Datasets can be in multiple containers
  - Containers can be in multiple containers





Interfaces

- Communication with Rucio:
  - Command Line Interface
    - User CLI
    - Admin CLI
  - Web User Interface
  - Python API
  - Rest API
- Documentation: https://rucio.readthedocs.io/en/latest/

Metadata

- Metadata are custom attributes on data identifiers
  - Support for arbitrary metadata has just been implemented
- Rucio supports different kinds of metadata
  - System-defined, e.g., size, checksum, did_type, is_open, created_at
  - Physics, e.g., number of events, GUID
  - Workflow management system, e.g., which task or job produced the file
  - Data management, necessary for the organisation of data
- Metadata provides another namespace
  - Datasets are searchable by name and metadata

Rucio Rules

- Replica management in Rucio is based on replication rules
- A replication rule defines the minimal number of replicas to be kept on a set of RSEs
  - e.g.: 2 replicas of file user.jdoe:file_001 on any TAPE system
- Declarative data management instead of imperative data management
  - “Two copies of X on TAPE” vs “Copy of X on TAPESYS_4 + Copy of X on TAPESYS_9”
  - Easier to use, optimize storage space, minimize number of transfers
- Multiple ownership of data
  - Rules not only invoke transfers, but also protect data from deletion

Site Configuration

Example: Using admin CLI to setup a new site

```
rucio-admin rse add MANCHESTER
rucio-admin rse set-attribute --rse MANCHESTER --key fts --value https://fts3-test.gridpp.rl.ac.uk:8446
rucio-admin rse set-attribute --rse MANCHESTER --key istape --value False
rucio-admin account set-limits root MANCHESTER 1000000000000000
```

Example: Using Python API to set protocols

```
from rucio.core.rse import add_protocol
tmp = {'hostname':'bohr3226.tier2.hep.manchester.ac.uk',
          'scheme':'gsiftp',
          'port':2811,
          'prefix':'/dpm/tier2.hep.manchester.ac.uk/home/skatelescope.eu/rucio',
          'impl':'rucio.rse.protocols.gfal.Default',
          'domains':{'lan':{'read':1, 'write':1, 'delete':1, 'third_party_copy':1}, 'wan':{'read':1, 'write':1, 'delete':1,
'third_party_copy':1}}}
add_protocol('MANCHESTER', parameter=tmp)
```

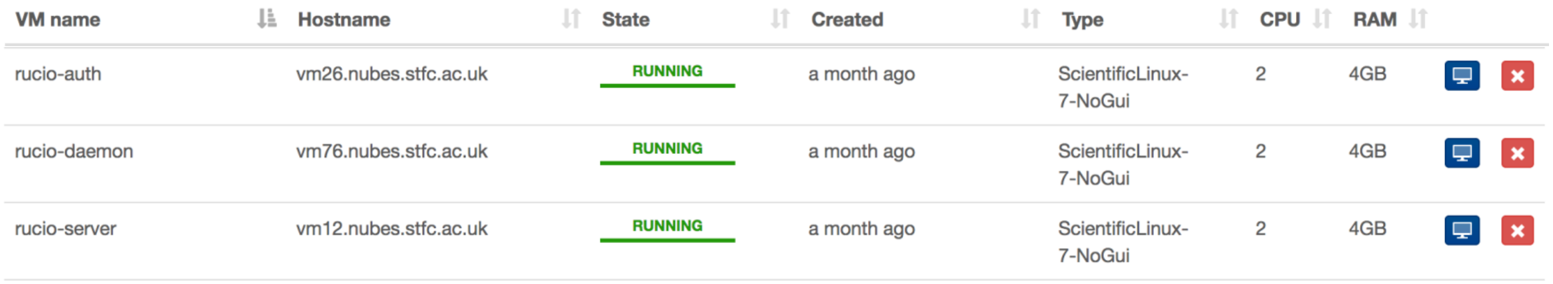
RAL Setup

Rucio at RAL

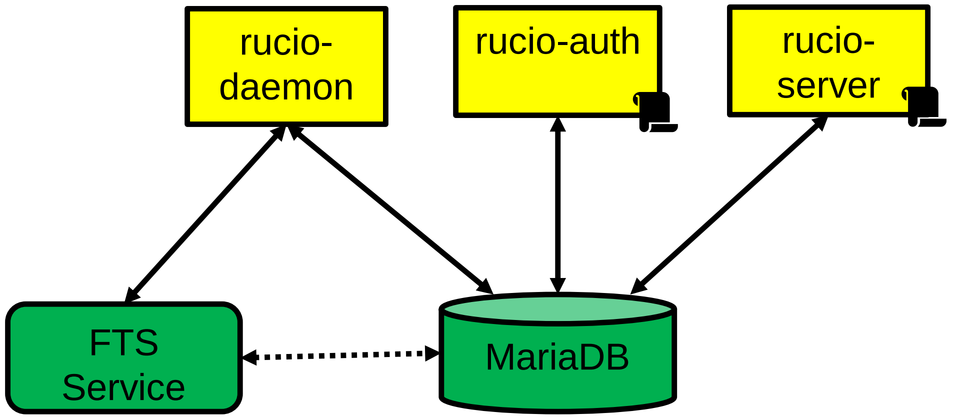
- I believe Rucio is the best way for the Tier-1 to integrate its storage services Castor (its replacement) and Echo.
  - Will be a key part of our GridPP6 bid.
- In April setup a Rucio instance for SKA for them to evaluate it.
- Since then have applied for money from IRIS to create a digital asset: “A Multi-VO Rucio instance”

SKA setup

Created VMs on Open Nebula Cloud



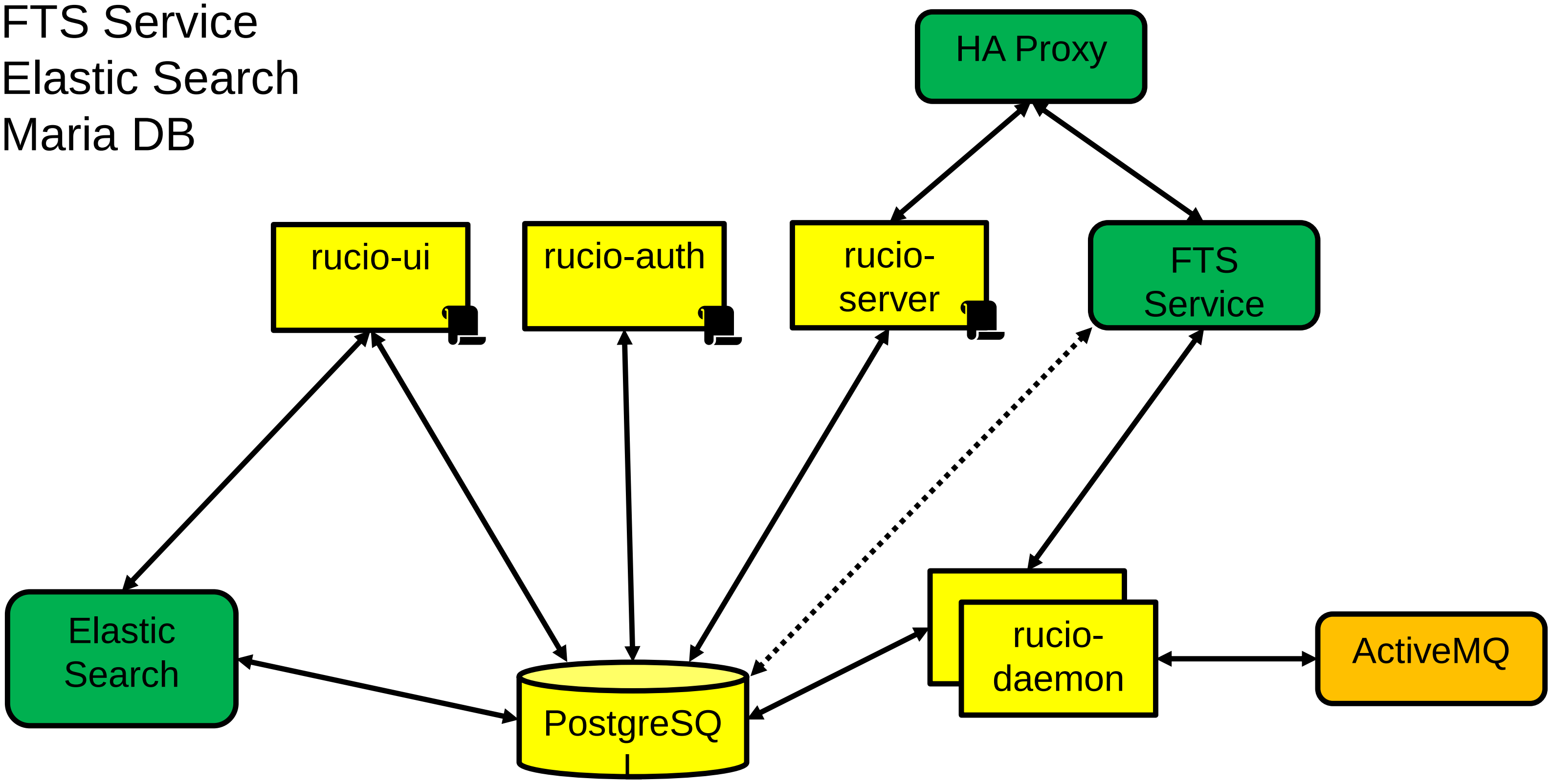
VM name	Hostname	State	Created	Type	CPU	RAM		
rucio-auth	vm26.nubes.stfc.ac.uk	RUNNING	a month ago	ScientificLinux-7-NoGui	2	4GB	🖥	✖
rucio-daemon	vm76.nubes.stfc.ac.uk	RUNNING	a month ago	ScientificLinux-7-NoGui	2	4GB	🖥	✖
rucio-server	vm12.nubes.stfc.ac.uk	RUNNING	a month ago	ScientificLinux-7-NoGui	2	4GB	🖥	✖



EU proposal to get funding for effort to run scale testing of SKA workflows.

Expanded Plans

Make use of existing services:
HA Proxy
FTS Service
Elastic Search
Maria DB



Rucio machines are all small VMs that will be setup on SCD Cloud infrastructure

Future work

- At RAL:
  - Move Rucio VMs to Openstack.
  - Setup dedicated database backend.
  - I have funding for development work to make a single Rucio instance able to support multiple VOs.
- HTCondor developers are looking at integration with Rucio.
- Both Imperial (GridPP DIRAC) and BNL (Belle2) are looking at Rucio integration with DIRAC.

Questions?