SKA Report on ESCAPE Data Lake Initial Trials

Data Injection Trial 1

Overview

In preparation for the initial report on the ESCAPE data lake prototyping, ESCAPE partners have been asked to perform some preliminary tests to evaluate the efficacy of the data lake technology stack for their respective use cases. Here we give a brief overview of the testing conducted by SKAO. So far, this campaign has been conducted at the Rucio (data lake orchestrator) level, using randomly generated files, rather than real or simulated data sets.

Environment

SKA testing has been performed using the *rucio-analysis* package, available on GitHub (https://github.com/ESCAPE-WP2/rucio-analysis). This offers a Python wrapper around the Rucio CLI, enabling tests to be written in Python and test parameters to be varied in associated configuration files. The tests are run in a Docker containerised environment, which requires an X509 certificate to be mounted in order to make calls to the Rucio REST server. Whilst test development can be performed on any machine with Docker, tests are executed on a dedicated cloud machine hosted by STFC Cloud. Full setup instructions are provided on the GitHub page.

Functional Testing

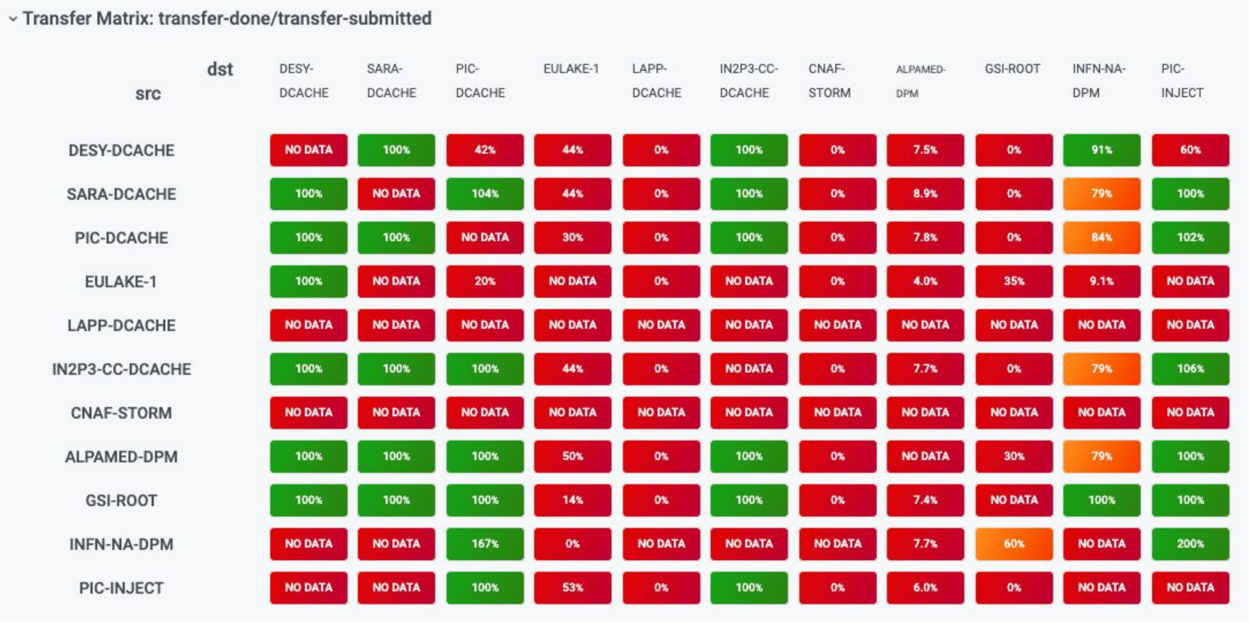
A functional test (*TestReplication*) of the full grid of sites runs every hour (minute 0) in the Rucio scope SKA_SKAO_BARNSLEY-testing. For each Rucio storage element (RSE), a single 100 kB file is uploaded, and then a replication rule for all other sites is added. This allows identification of sites for which file transfers are currently broken.

Load Testing

In addition to functional testing, a load test has been created (*TestReplicationBulk*), which also runs hourly (minute 10), to test higher velocity data upload and replication. Currently this uploads 1600 15 kB files to the DESY-DCACHE RSE, replicating to SARA-DCACHE and EULAKE-1, in the Rucio scope SKA_SKAO_COLLINSON. The test run time is approx 15 min and increases with the number of files. Thus, while it would be possible to increase the scale of this test by a small factor, a larger scale increase could risk the test running for longer than the job interval (i.e. an hourly job taking more than an hour to run). To mitigate this would require the test to be run less frequently, or a compute instance with more cores being provisioned for the task.

Results

Results can be viewed in the ESCAPE Rucio Events Grafana dashboard. Transfer success ratios are at 100% for the load test (which tests only 3 sites) and typically as follows for the functional test:



Transfer Matrix: transfer-done/transfer-submitted

src \ dst	DESY-DCACHE	SARA-DCACHE	PIC-DCACHE	EULAKE-1	LAPP-DCACHE	IN2P3-CC-DCACHE	CNAF-STORM	ALPAMED-DPM	GSI-ROOT	INFN-NA-DPM	PIC-INJECT
DESY-DCACHE	NO DATA	100%	42%	44%	0%	100%	0%	7.5%	0%	91%	60%
SARA-DCACHE	100%	NO DATA	104%	44%	0%	100%	0%	8.9%	0%	79%	100%
PIC-DCACHE	100%	100%	NO DATA	30%	0%	100%	0%	7.8%	0%	84%	102%
EULAKE-1	100%	NO DATA	20%	NO DATA	0%	NO DATA	0%	4.0%	35%	9.1%	NO DATA
LAPP-DCACHE	NO DATA	NO DATA	NO DATA	NO DATA	NO DATA	NO DATA	NO DATA	NO DATA	NO DATA	NO DATA	NO DATA
IN2P3-CC-DCACHE	100%	100%	100%	44%	0%	NO DATA	0%	7.7%	0%	79%	106%
CNAF-STORM	NO DATA	NO DATA	NO DATA	NO DATA	NO DATA	NO DATA	NO DATA	NO DATA	NO DATA	NO DATA	NO DATA
ALPAMED-DPM	100%	100%	100%	50%	0%	100%	0%	NO DATA	30%	79%	100%
GSI-ROOT	100%	100%	100%	14%	0%	100%	0%	7.4%	NO DATA	100%	100%
INFN-NA-DPM	NO DATA	NO DATA	167%	0%	NO DATA	NO DATA	NO DATA	7.7%	60%	NO DATA	200%
PIC-INJECT	NO DATA	NO DATA	100%	53%	0%	100%	0%	6.0%	0%	NO DATA	NO DATA

Note: some sites show values with greater than 100% success ratios - this is due to requests submitted **outside** of the graphing time interval completing **within** the time interval. This issue is not addressable with the current Grafana/ES combination.

Next steps

The next steps in this data lake testing campaign will be to increase the coverage of functionality we test, including file download and deletion, as well as creating replication rules based on a target quality-of-service, rather than site.

Data Injection Trial 2

Overview

Largely QoS based testing done during this round including building a test as a QoS health check ensuring all expected levels of QoS are available and data lifecycle test simulating SKA data flow based on QoS.

QoS testing

QoS Health check

This test uploads and replicates data to all QoS labels available as a way to perform a QoS health check. This test (along with the replication tests) would run regularly at a frequency of once per day to ensure that there is at least one RSE at all time available for all the QoS labels.

Simulate SKA data lifecycle

This test takes input parameters in the form of an ordered list of QoS labels and a total data lifecycle. Test file is uploaded to an RSE with the first QoS label, and rules of increasing lifetimes created for the remainder of the QoS labels. Currently, test file is uploaded to OPPORTUNISTIC storage for 25% of the total data lifecycle (say, T), replicated to FAST for 50%T, CHEAP-ANALYSIS for 75%T and SAFE for the entire data lifecycle T. Lastly, the file is downloaded and the local adler32 checksum is validated against the remote checksum.

An example of one such test file shown below where QoS based rules exist with varying lifetimes.

```
[root@dddac6bb20fa src]# rucio list-rules SKA_SKAO_JOSHI-testing:14KB_141020T08.29.17
ID                                ACCOUNT    SCOPE:NAME                                  STATE[OK/REPL/STUCK]
RSE_EXPRESSION        COPIES  EXPIRES (UTC)        CREATED (UTC)
--------------------------------  ---------  ------------------------------------------  ----------------------
------------------  --------  -------------------  -------------------
a9208f7ace924e988980aed641c5f28b  rjoshi     SKA_SKAO_JOSHI-testing:14KB_141020T08.29.17  OK[1/0/0]
QOS=SAFE                   1  2020-10-15 08:29:26  2020-10-14 08:29:26
c17518d7e8a54bc9a430d3d81529e2c5  rjoshi     SKA_SKAO_JOSHI-testing:14KB_141020T08.29.17  OK[1/0/0]
QOS=FAST                   1  2020-10-14 20:29:25  2020-10-14 08:29:25
82e2430224ca4fe5b9727f084178431d  rjoshi     SKA_SKAO_JOSHI-testing:14KB_141020T08.29.17  OK[1/0/0]
QOS=CHEAP-ANALYSIS         1  2020-10-15 02:29:25  2020-10-14 08:29:25
```

The rucio commands to list rules, list rules history, list dids in a dataset, rule info and list file replicas were run manually outside of the testing framework.

The data lifecycle test was also run to replicate to a destination RSE based on QoS from a source RSE based on QoS (ie rules of the form replicate from QOS=OPPORTUNISTIC to QOS=FAST).

Next steps

Workflow integration - Download Rucio managed data, run a workflow on it and upload the results. We would look to do this in the form of Jupyter notebooks.

Data Injection Trial 3

Overview

The primary task in this trial was to access the data in the datalake in some meaningful way. As part of the data lifecycle QoS tests in trial 2, downloading data and performing checksums has already been demonstrated. This was done in the rucio-analysis framework hosted on github.
In addition to this, we have performed datalake access and interaction with the data from the Rucio JupyterLab extension running on the ESCAPE JupyerHub instance. We downloaded data from within the Rucio environment and ran a Jupyter notebook to visualise a FITS file.

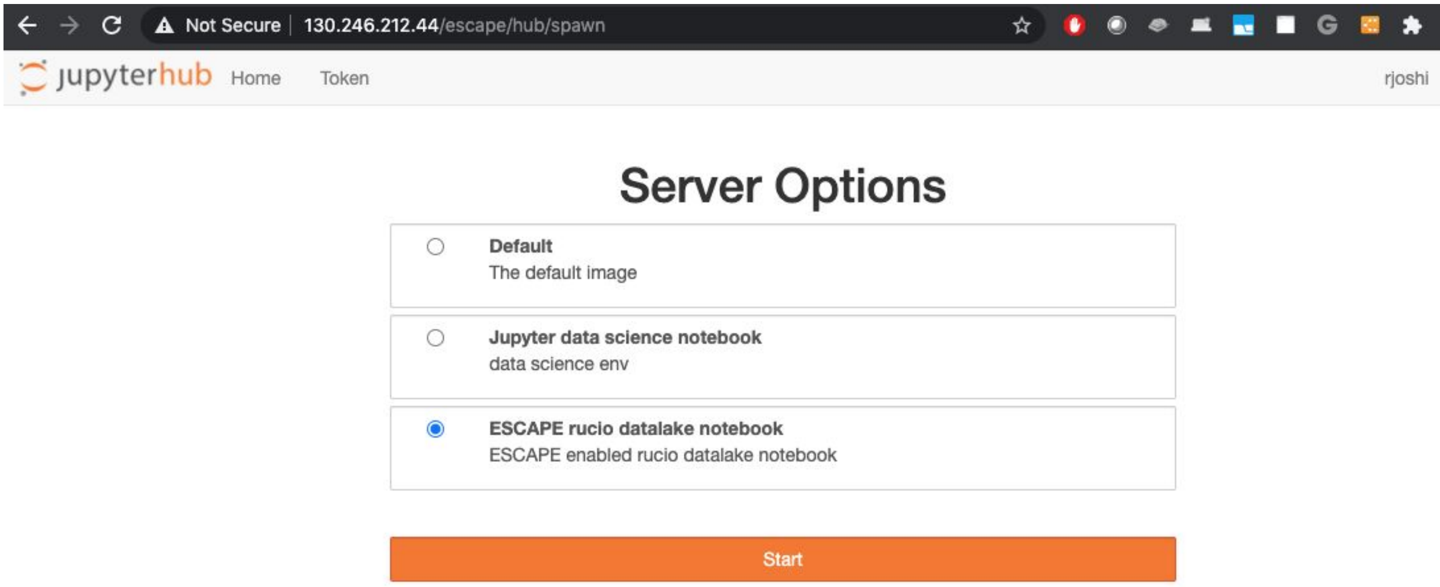
Description

- FITS file uploaded to data lake with a lifetime of 1 week to DESY-DCACHE (SKA_SKAO_JOSHI-testing:B1_1000h_0.3tr.fits). This file contains simulated radio sources.

```
[root@624071128b1d src]# rucio upload --rse DESY-DCACHE --lifetime 604800 --scope
SKA_SKAO_JOSHI-testing --register-after-upload /opt/rucio/etc/B1_1000h_0.3tr.fits
2020-10-26 15:28:26,489 INFO    Preparing upload for file B1_1000h_0.3tr.fits
2020-10-26 15:28:29,491 INFO    Trying upload with davs to DESY-DCACHE
2020-10-26 15:28:35,124 INFO    Successful upload of temporary file.
davs://dcache-door-doma01.desy.de:2880//escape/wp2_rucio_testbed/desy_dcache/SKA_SKAO_J
OSHI-testing/3f/f9/B1_1000h_0.3tr.fits.rucio.upload
2020-10-26 15:28:35,746 INFO    Successfully uploaded file B1_1000h_0.3tr.fits
2020-10-26 15:28:36,673 INFO    Successfully added replica in Rucio catalogue at
```

```
DESY-DCACHE
2020-10-26 15:28:36,792 INFO     Successfully added replication rule at DESY-DCACHE
```

- Spin up a Rucio user environment on ESCAPE JupyterHub instance



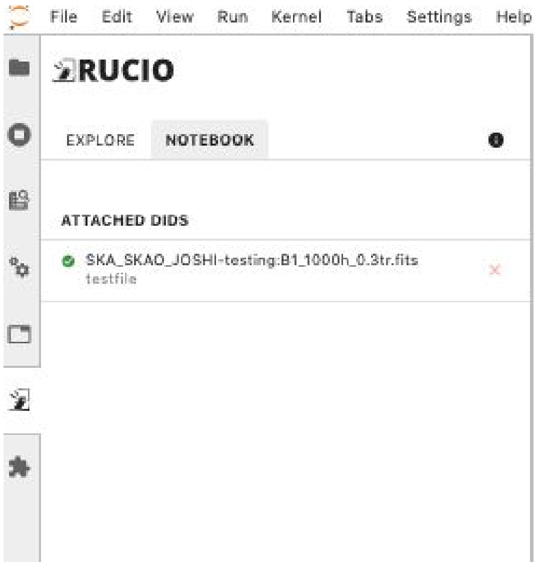
Server Options

○ **Default**
The default image

○ **Jupyter data science notebook**
data science env

● **ESCAPE rucio datalake notebook**
ESCAPE enabled rucio datalake notebook

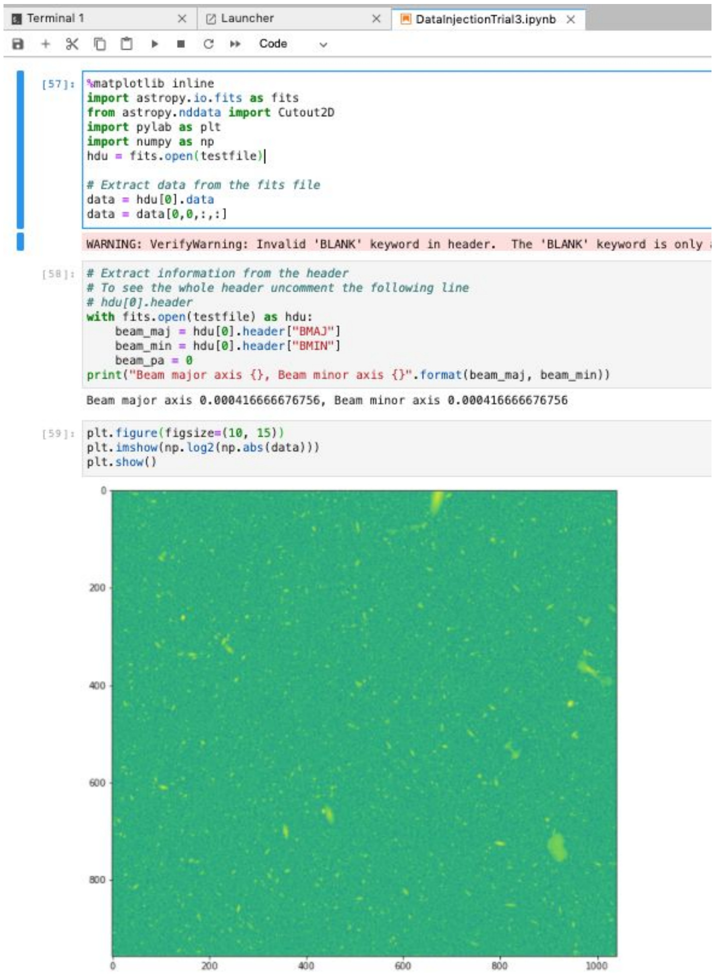
- Find, download and add to the notebook as testfile



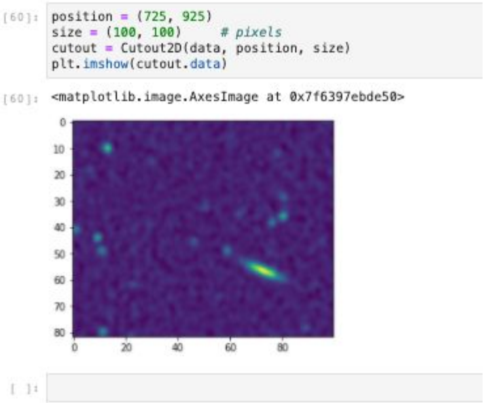
ATTACHED DIDS

SKA_SKAO_JOSHI-testing:B1_1000h_0.3tr.fits
testfile

- Run a notebook to visualise the FITS file, see header information and perform a cutout on the file



```
[57]: %matplotlib inline
      import astropy.io.fits as fits
      from astropy.nddata import Cutout2D
      import pylab as plt
      import numpy as np
      hdu = fits.open(testfile)

      # Extract data from the fits file
      data = hdu[0].data
      data = data[0,0,:,:]

      WARNING: VerifyWarning: Invalid 'BLANK' keyword in header.  The 'BLANK' keyword is only

[58]: # Extract information from the header
      # To see the whole header uncomment the following line
      # hdu[0].header
      with fits.open(testfile) as hdu:
          beam_maj = hdu[0].header["BMAJ"]
          beam_min = hdu[0].header["BMIN"]
          beam_pa = 0
      print("Beam major axis {}, Beam minor axis {}".format(beam_maj, beam_min))

      Beam major axis 0.000416666676756, Beam minor axis 0.000416666676756

[59]: plt.figure(figsize=(10, 15))
      plt.imshow(np.log2(np.abs(data)))
      plt.show()
```



```
[60]: position = (725, 925)
      size = (100, 100)     # pixels
      cutout = Cutout2D(data, position, size)
      plt.imshow(cutout.data)

[60]: <matplotlib.image.AxesImage at 0x7f6397ebde50>

[ ]:
```

- Notebook uploaded to data lake for anyone who wants to run it

```
jovyan@jupyter-rjoshi:~$ rucio upload --rse DESY-DCACHE --lifetime 604800 --scope
SKA_SKAO_JOSHI-testing --register-after-upload DataInjectionTrial3.ipynb
2020-10-26 18:04:41,753 INFO    Preparing upload for file DataInjectionTrial3.ipynb
2020-10-26 18:04:42,219 INFO    Trying upload with davs to DESY-DCACHE
2020-10-26 18:04:43,309 INFO    Successful upload of temporary file.
davs://dcache-door-doma01.desy.de:2880//escape/wp2_rucio_testbed/desy_dcache/SKA_SKAO_J
OSHI-testing/21/63/DataInjectionTrial3.ipynb.rucio.upload
2020-10-26 18:04:43,654 INFO    Successfully uploaded file DataInjectionTrial3.ipynb
2020-10-26 18:04:43,898 INFO    Successfully added replica in Rucio catalogue at
DESY-DCACHE
2020-10-26 18:04:43,989 INFO    Successfully added replication rule at DESY-DCACHE
```