



D2.2.5 Design for ingestion of User-generated Products

WP2.2: Data Ingestion and Publication

Project Acronym LUSC-B
Project Title UK Involvement in the Large Synoptic Survey Telescope
Document Number LUSC-B-35

Submission date	21/JUL/23
Version	1.1
Status	Final
Author(s) inc. institutional affiliation	Mike Read (Edinburgh), George Beckett (Edinburgh)
Reviewer(s)	Tom Wilson (Exeter), Aaron Watkins (Herts)

Dissemination level	
Public	

Version History

Version	Date	Comments, Changes, Status	Authors, contributors, reviewers
0.1	5/DEC/22	Initial document skeleton	MGB
1.0	4/JUL/23	Full draft	MAR/MGB
1.1	8/AUG/23	Final version addressing review comments	MAR

Table of Contents

VERSION HISTORY	2
1 EXECUTIVE SUMMARY	5
2 INTRODUCTION	6
2.1 PURPOSE.....	6
2.2 THE WORK-PACKAGES AND THEIR DATASETS	6
2.2.1 <i>WP3.5 - LSST and near-infrared data fusion</i>	6
2.2.2 <i>WP3.11 - Cross matching and astrometry at LSST depths (WP3.3 in Phase C)</i>	6
2.3 GLOSSARY OF ACRONYMS.....	6
3 THE DATA SYSTEM	8
3.1 THE DATA BUTLER REPOSITORY	8
3.2 QSERV DATABASE.....	8
4 WORKFLOW - LSST AND NEAR-INFRARED DATA FUSION	9
4.1 DATA PREPARATION	9
4.2 DATA TRANSFER.....	9
4.3 BUTLER INGEST.....	9
4.4 QSERV INGEST	10
5 WORKFLOW - CROSS MATCHING AND ASTROMETRY AT LSST DEPTHS	12
5.1 DATA PREPARATION	12
5.2 DATA TRANSFER.....	12
5.3 QSERV INGEST	12
6 PUBLISHING TO THE RSP	13
7 USE CASES	14
7.1 SXDS REGION, NEAR-INFRARED DATA FUSION	14
7.1.1 <i>Issues with the infra-red fusion data ingestion</i>	15
7.2 GAIADR3 – CATWISE CROSS-MATCH	15
7.2.1 <i>Pre-ingestion work</i>	15
7.2.2 <i>Ingestion and publication</i>	15
7.2.3 <i>Issues with the ingesting gaia-catwise data</i>	16
8 NEXT STEPS	17
8.1 NEAR INFRARED DATA FUSION	17
8.2 CROSS-MATCH	17
9 CONCLUSION	18
10 REFERENCES	19

Index of Figures

Figure 1 - a screenshot of the Qserv dashboard during data ingestion of the VIDEO SXDS region..... 10

Figure 2 - a screenshot of a notebook accessing the SXDS butler repository. 14

D2.2.5 DESIGN FOR INGESTION OF USER-GENERATED PRODUCTS

Figure 3 - screenshot showing the number of rows ingested into the all-sky crossmatch database (rows in chunks, rows in chunk overlaps, total rows. 15

Figure 4 - screenshot of an example notebook querying the crossmatch TAP service 16

1 Executive Summary

In addition to the main LSST survey products and external ancillary survey datasets, the UK-DAC will host and serve data products generated by some of the UK work-packages. Currently there are two such work-packages, WP3.11 (Cross matching and astrometry at LSST depths) and WP3.5 (LSST and near-infrared data fusion). The same infrastructure used for the LSST products is to be utilised, namely Qserv for the relational database and the data butler for the flat file products (images etc). The data will be accessed via the Rubin Science Platform. This document gives an overview of the steps involved in publishing data from these two work-packages and the next steps for this work.

2 Introduction

2.1 Purpose

Alongside the main LSST data releases the UK-DAC will host and serve datasets generated by some of the UK work-packages, to facilitate the science goals of the UK community. Here we describe the workflow for obtaining, preparing, ingesting, and publishing these data.

2.2 The work-packages and their datasets

The UK-DAC is currently working with WP3.11 (Cross matching and astrometry at LSST depths) and WP3.5 (LSST and near-infrared data fusion). This interaction is described in detail in the interface definition documents for each work-package [1][2].

A summary of these two work-packages and the data they produce is given below.

2.2.1 WP3.5 - LSST and near-infrared data fusion (WP3.2 in Phase C)

This work-package is using the LSST pipeline to process optical images alongside near-infrared images from VISTA. Ahead of LSST observations the optical images are from the Hyper Suprime-Cam (HSC) survey. The near-infrared data comes from the VIDEO, VIKING and VHS, VISTA surveys.

The pipeline is currently being developed and run on CSD3 where it produces images of various types at various stages and generates catalogues from the final calibrated co-adds. The pipeline is currently described in a preprint [3]. Data in up to ten wavebands are produced, g, r, i, z and y for HSC and Z, Y, J, H and Ks for VISTA. Two types of final single band catalogues are generated, meas(ured) and forced. With the meas catalogues being suited to studies of object properties in a specific filter and the forced catalogues for investigations using colours. The most recent run of the pipeline has also outputted a band-merged catalogue containing the most useful attributes across all available wavebands.

Full production runs of this work-package will generate petabytes of data across millions of files.

2.2.2 WP3.11 - Cross matching and astrometry at LSST depths (WP3.3 in Phase C)

This work-package is providing state-of-the-art cross-matching algorithms (Macauff) which include the effects of crowding. Models of astrometric uncertainties and photometric information, together with proper motions, where available, are used to determine the best cross-match between two surveys [4].

Ultimately cross-matches are to be carried out against the main object catalogue from LSST data releases and a number of external surveys (see preliminary cross-match list [5]). Currently the Gaia and WISE datasets are being used to develop the algorithms and system. The Gaia-WISE cross-matched data are also scientifically useful in their own right and provide a good use case for the Rubin Science Platform (RSP). As for WP3.5, full cross-matches are also currently run on CSD3.

The catalogue data volume generated by this work-page in production is estimated to be tens of terabytes.

2.3 Glossary of Acronyms

CSV – Comma-Separated Values

D2.2.5 DESIGN FOR INGESTION OF USER-GENERATED PRODUCTS

HSC – Hyper Suprime-Cam

MSSQL – Microsoft SQL server

MySQL – My Structured Query Language

NGP – North Galactic Plane

RAL – Rutherford Appleton Labs

RSP – Rubin Science Platform

SXDS – Subaru/XMM-Newton Deep Survey

TAP – Table Access Protocol

TSV – Tab-Separated Values

VIDEO – VISTA Deep Extragalactic Observations

VISTA – Visible and Infrared Survey Telescope for Astronomy

WFAU – Wide-field Astronomy Unit

YAML – Yet Another Markup Language

3 The data system

The output of the work-packages described mainly comprises image and catalogue files. Once transferred to the UK-DAC these will be hosted in either a data butler repository [6] and/or the Qserv database system [7]. Users will access the data via the RSP or Table Access Protocol (TAP) service.

3.1 The data butler repository

The image and catalogue files (e.g., FITS files) generated by WP3.5 will be stored and accessed via a butler repository. Metadata from the files are held in the butler's registry database together with the file path/URL. The butler repository can consist of various datasets and collections of files – e.g., raws, single-visits, coadds, and skymaps.

The butler's registry database can be based on an instance of SQLite but larger registries will be stored in PostgreSQL. The WP3.5 butler repository on CSD3 currently uses SQLite.

3.2 Qserv database

Catalogue data will be hosted in the Qserv relational database system. Based around MySQL instances, Qserv, at scale, is distributed across tens of nodes, with each node hosting sections of the data, and employs shared scanning techniques to enable fast query times on exceptionally large datasets. Catalogues generated by WP3.5 and WP3.11 will be ingested into Qserv.

Our current Qserv system is a Kubernetes deployed cluster, hosted on the Somerville OpenStack system. It currently has five worker nodes.

4 Workflow - LSST and near-infrared data fusion

4.1 Data preparation

The modified LSST pipeline, run at CSD3, outputs images as FITS files and catalogues in FITS or parquet format.

These flat file products are all registered in a butler repository and stored in various collections.

In addition to the catalogue files recorded in the butler, the main science catalogues, from the deep calibrated coadded data collection, are to be ingested into the Qserv relational database. Hosting the main science catalogues in Qserv allows for more powerful and flexible querying.

Currently the Qserv partitioning and ingest software requires delimited ASCII files (e.g., tab-separated or comma separated values TSV/CSV). So, in addition to the files produced by the pipeline and recorded in the butler at CSD3, CSV files are also generated by the Dev team.

Given the large data volume that will be generated by this work-package it has been decided to only transfer and ingest the (deep) coadd calexp images, catalogues and their skymaps to the Edinburgh UK-DAC. These are the most useful products for science, the other data products will be transferred to Echo S3 object storage at RAL and ultimately available from a butler repository pointing to file endpoints there. If we discover later that users frequently require access to collections other than the deep coadd calexp and accessing them via the Echo S3 butler is not performant than we will re-visit this decision.

A list of deep coadd calexp files to transfer to the butler on the UK-DAC is generated from the butler on CSD3. This is used to drive the butler file transfer.

4.2 Data transfer

To date file transfers to the UK-DAC have been carried out using either rsync or scp. These have coped well with the data volumes encountered (the largest transfer being some 40TB) but as the transfers grow, it is likely that another technology will be needed. This is to be investigated. The bulk transfers to Echo storage at RAL are to be carried out using the Rucio distributed data management system. The overheads associated with running the Rucio service mean it's unlikely to be adopted for the transfers to the Edinburgh UK-DAC.

The files destined for the UK-DAC's butler and the CSV files for the Qserv database are transferred and handled separately.

It is thought that some combination of gridFTP and XrootD technologies can be employed to handle the transfers to the UK-DAC as data volume grows.

4.3 Butler ingest

Once transferred to the UK-DAC, the image and catalogue flat files generated by the CSD3 pipeline are to be imported to a local butler repository. This is done using a YAML file created from the CSD3 butler.

https://github.com/lstt-uk/lstt-ir-fusion/blob/master/dmu4/dmu4_VIDEO/slurm/make_export.py.

D2.2.5 DESIGN FOR INGESTION OF USER-GENERATED PRODUCTS

```
REPO="/path-to-repo/butler-repo"  
butler = dafButler.Butler(REPO,writeable=True)  
butler.import_(  
    directory='/path-to-repo/butler-  
repo/dmu4/dmu4_VIDEO/data',  
    filename='/path-to-repo/butler-  
repo/dmu4/dmu4_VIDEO/slurm/exports/exports_20230207.yaml'  
)
```

At CSD3 the butler uses SQLite as the registry database but in the UK-DAC. the butler is now set-up to use PostgreSQL as this is the recommendation for larger repositories.

4.4 Qserv ingest

The band merged catalogues and single band meas/forced catalogues generated as CSV files are to be loaded into Qserv.

Prior to ingestion into Qserv the transferred CSV files are first checked for consistency. Qserv has the concept of director and child tables. Here the director table is to be the band-merged output and the child tables are the individual band meas/forced catalogues. The catalogues are organised by sky tract and patch. Each child catalogue file must have the same number of rows as the corresponding director catalogue to ensure a one-to-one correlation between the objectIds of the sources. The pipeline should guarantee this, but output sometimes fails or is truncated. Single-band catalogues with the incorrect number of rows are not passed on for Qserv ingestion

To distribute data across the Qserv worker nodes it first needs to be partitioned into chunks (regions on the sky). The validated CSV catalogue files are run through the sph-partition [8] application which creates a set of chunk CSV files for each table. The qserv-ingest tool [9] is then used to load these files into a Qserv database.

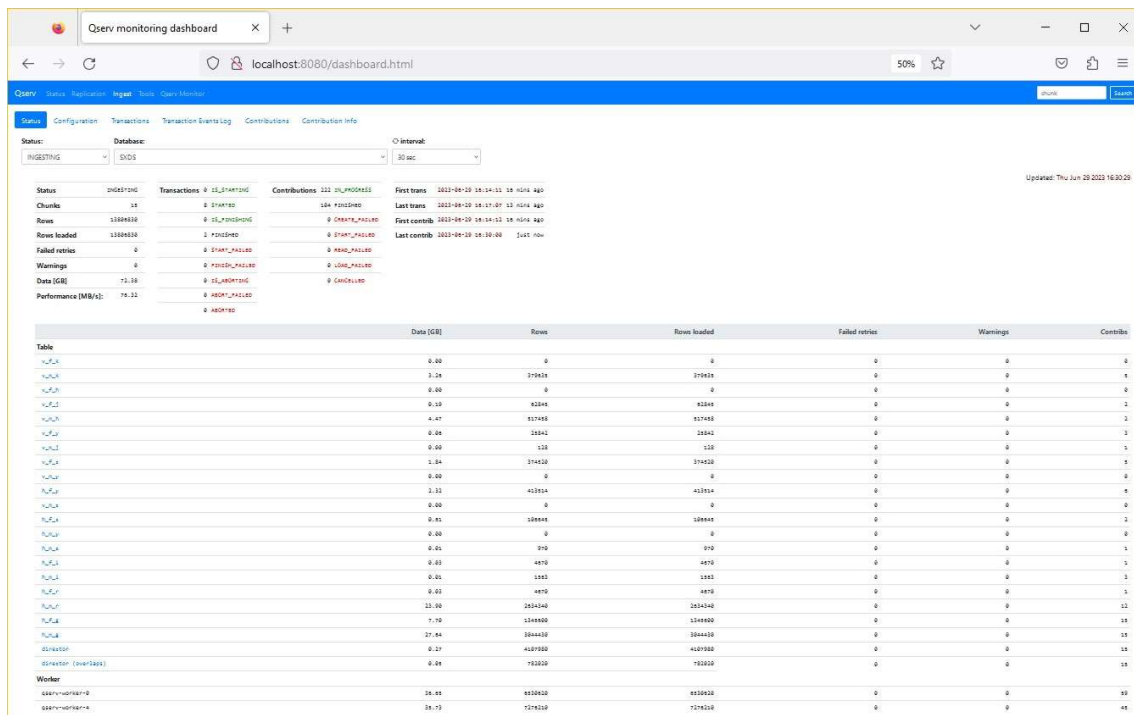


Figure 1 - a screenshot of the Qserv dashboard during data ingestion of the VIDEO SXDS region

D2.2.5 DESIGN FOR INGESTION OF USER-GENERATED PRODUCTS

The ingest is driven by a series of JSON files describing the database schema and location of the chunked data files. Typically, the files are located on a webserver and accessed by `qserv-ingest` over HTTP. The loading of data into Qserv is discussed in more detail in the Confluence document, *WP3.5 - Notes on ingesting SXDS into Qserv*, [10]

5 Workflow - Cross matching and astrometry at LSST depths

5.1 Data preparation

The UK-DAC

Prior to cross-matching, skinny flagged catalogues of the two surveys being paired are generated by the UK-DAC in collaboration with the work-package team. These catalogues contain the minimum attributes required for the cross-match: IDs, magnitudes, quality flag, optimum waveband, etc. Sources flagged as being poor quality in all wavebands for a given survey are removed from these input catalogues.

The input catalogues are then partitioned on the sky (using the *sph-partition* tool for convenience) and transferred to CSD3 as a set of chunked CSV files.

The partitioning of the data allows for the cross-matching to be performed on a chunk-by-chunk basis and for the matching of separate chunks to be run in parallel.

The output from the cross-matching are sets of chunk CSV files. One set contains the sources that have been cross-matched. The other two sets contain the unmatched data from each of the surveys.

5.2 Data transfer

The data volume of the cross-matched results is relatively modest and the sets of chunked data are readily transferred to the UK-DAC using *rsync*.

5.3 Qserv ingest

Prior to ingest the cross-match output is checked to see that the number of sources present in the matched and unmatched files is the same as the inputs.

Each set of CSV files is to be ingested into a separate table in a Qserv database. Each of these matched and unmatched tables needs to be partitioned prior to ingestion. Here the partitioned scheme (region size) will be different to that used for the input files. This is because it must be the same as the partitioning of the surveys being matched so that the cross-match output is placed on the same Qserv worker nodes as the underlying surveys.

The partitioned data is again ingested using *qserv-ingest*.

6 Publishing to the RSP

User access to the output from the work-packages discussed here will be via the Rubin Science Platform (RSP).

To achieve this the butler repository location needs to be visible and accessible to the RSP compute nodes e.g. via a shared Ceph volume. Credentials then need to be configured to allow the RSP to connect to the PostgreSQL registry.

To publish a Qserv database a Felis [11] YAML file describing the database schema (table and column names and types) is generated and registered to the RSP so that it can build the TAP service and interface.

7 Use cases

7.1 SXDS region, near-infrared data fusion

The LSST and near-infrared data fusion pipeline was run at CSD3 on the SXDS sky region. The optical data coming from HSC and the infrared data from VIDEO. The deep coadds and CSV catalogue data were transferred to the UK-DAC using rsync. The data volume was around 3TB.

A butler repository was created using a PostgreSQL registry and loaded with the coadd files.

The CSV files were cleaned (incomplete tracts/patches being removed), partitioned and ingested into Qserv. The director table and ten child tables each contained 5 million rows.

The RSP was configured to access the butler repository and the Qserv TAP service. Notebooks were created on the RSP that queried the butler repository and the published database.

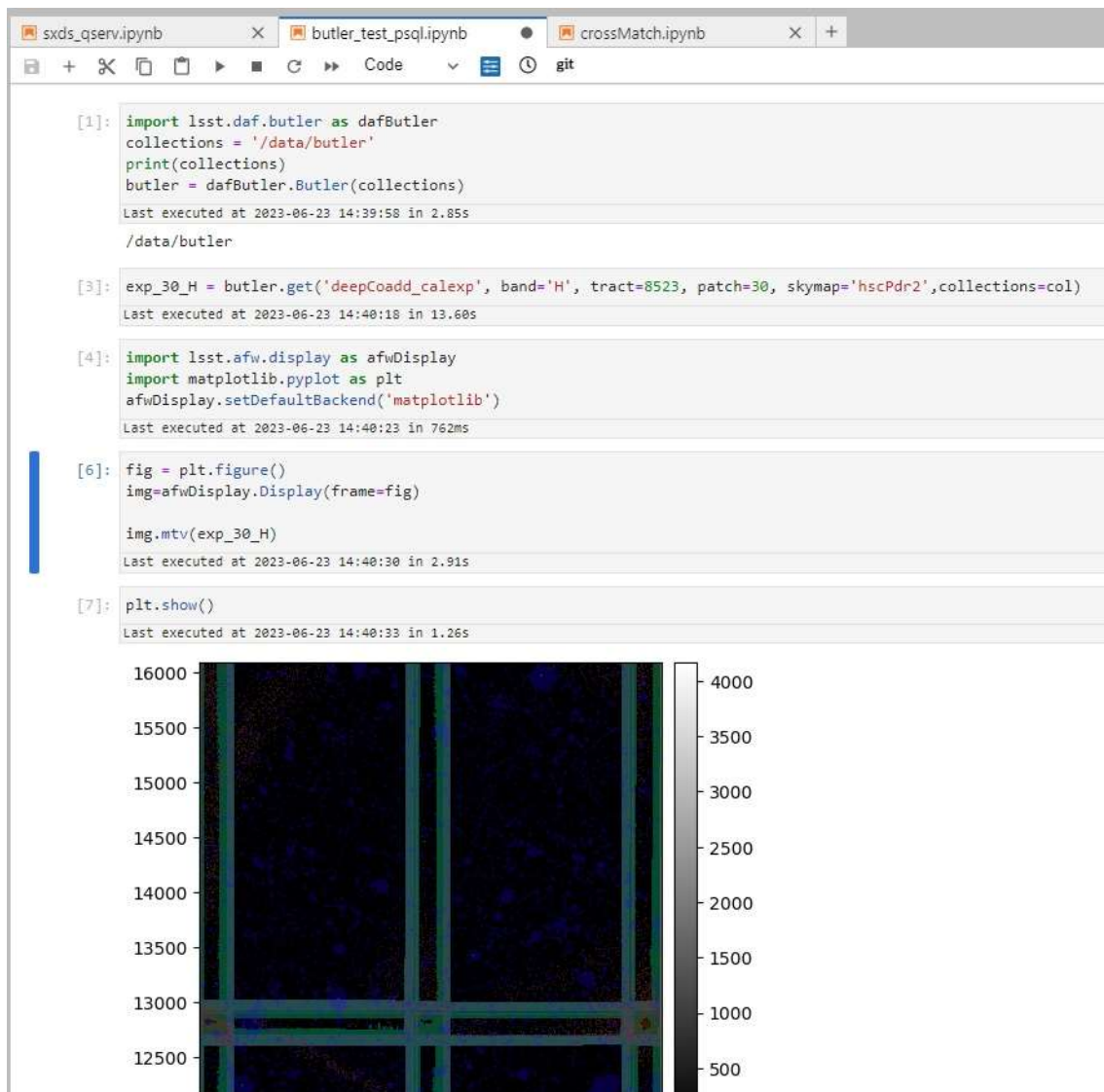


Figure 2 - a screenshot of a notebook accessing the SXDS butler repository.

7.1.1 Issues with the infra-red fusion data ingestion

Currently to access the PostgreSQL butler via the RSP, credentials need to be copied to the home directory of each user. A solution for this is being investigated.

The SXDS CSV data output by the pipeline contained some regions with inconsistent row counts across the individual wavebands and there were some regions with missing catalogues. This run did not produce the band merged catalogues so the director table was formed at the UK-DAC from a simple set of 3 attributes (objectId, RA and Dec). There were also a few minor issues with column names and the coordinates needed converting from radians to degrees. The latest pipeline run, awaiting transfer and ingestion, is thought to have addressed all the issues encountered.

7.2 GaiaDR3 – CatWISE cross-match

Skinny, flagged GaiaDR3 and CatWISE catalogues were generated at the UK-DAC from the full all-sky survey datasets [12]. These were then partitioned into some 1558 chunks. The chunks had a small overlap region to allow for cross-matching around the chunk edges. The chunks, totalling around 0.6TB, were transferred to CSD3 via rsync.

Following cross-matching at CSD3, the chunked results files were copied up to the UK-DAC.

7.2.1 Pre-ingestion work

Qserv should now be able to support joins between two director tables via a cross-match table. However only having moved to the current version of Qserv very recently we have not had the opportunity to investigate this. To enable user access to the full cross-matched data on a shorter timescale we pre-joined the output from cross-match process with the Gaia and CatWISE data.

The pre-ingestion checks confirmed what had already been relayed by the work-package team, namely that duplicates were present, and some input sources were missing from the results.

The main index on Qserv director tables has to be unique so the duplicates were removed, and the cross-match with the greatest probability for a given source retained.

7.2.2 Ingestion and publication

The pre-joined data were then partitioned and ingested using `qserv-ingest`. Initially the NGP region was ingested as this was the priority but subsequently the all-sky, matched and unmatched data were ingested into Qserv.

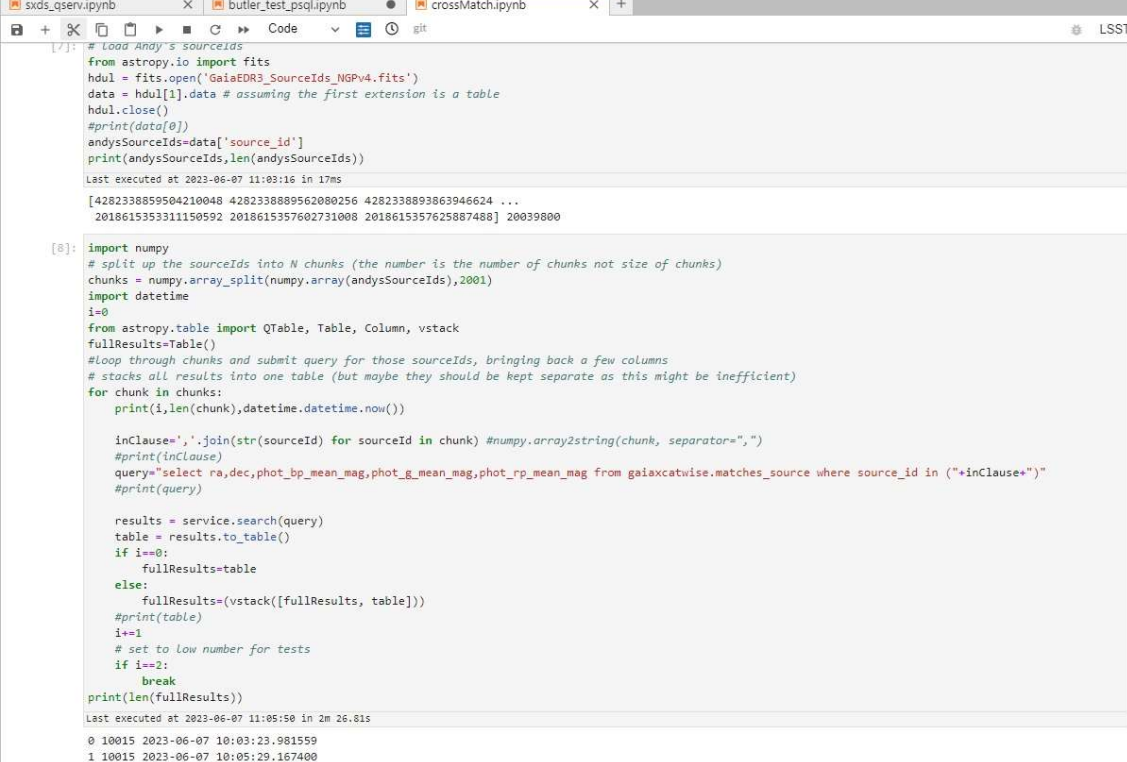
The all-sky `gaiaxcatwise` Qserv database holds around 750 million cross-matched sources in the `matches_source` table and around 1 billion sources in each of the unmatched tables.

gaiaxcatwiseallsky	matches_source	773780847	30802402	804583249
	gaia_nonmatches	978488818	38920900	1017409718
	catwise_nonmatches	1105128143	44003731	1149131874

Figure 3 - screenshot showing the number of rows ingested into the all-sky crossmatch database (rows in chunks, rows in chunk overlaps, total rows).

Following ingestion, the Felis YAML file was generated and used to publish the TAP service on the RSP.

A notebook was written to show how to query the cross-matched data and extract results for a large set of sources.



```

[7]: # Load Andy's sourceIds
from astropy.io import fits
hdl = fits.open('GaiaEDR3_SourceIds_NGPv4.fits')
data = hdl[1].data # assuming the first extension is a table
hdl.close()
#print(data[0])
andysSourceIds=data['source_id']
print(andysSourceIds,len(andysSourceIds))

Last executed at 2023-06-07 11:03:16 in 17ms

[4282338859504210048 4282338889562080256 4282338893863946624 ...
 2018615353311150592 2018615357602731008 2018615357625887488] 20039800

[8]: import numpy
# split up the sourceIds into N chunks (the number is the number of chunks not size of chunks)
chunks = numpy.array_split(numpy.array(andysSourceIds),2001)
import datetime
i=0
from astropy.table import QTable, Table, Column, vstack
fullResults=Table()
# Loop through chunks and submit query for those sourceIds, bringing back a few columns
# stacks all results into one table (but maybe they should be kept separate as this might be inefficient)
for chunk in chunks:
    print(i,len(chunk),datetime.datetime.now())

    inClause=', '.join(str(sourceId) for sourceId in chunk) #numpy.array2string(chunk, separator=",")
    #print(inClause)
    query="select ra,dec,phot_bp_mean_mag,phot_g_mean_mag,phot_rp_mean_mag from gaiaecatwise.matches_source where source_id in (" +inClause+ ")"
    #print(query)

    results = service.search(query)
    table = results.to_table()
    if i==0:
        fullResults=table
    else:
        fullResults=(vstack([fullResults, table]))
    #print(table)
    i+=1
    # set to low number for tests
    if i==2:
        break
print(len(fullResults))

Last executed at 2023-06-07 11:05:50 in 2m 26.81s

0 10015 2023-06-07 10:03:23.981559
1 10015 2023-06-07 10:05:29.167400

```

Figure 4 - screenshot of an example notebook querying the crossmatch TAP service

Two Exeter students, Andy Wilson and George Foley, adapted this notebook and successfully extracted the data they needed from the published cross-matched data using the RSP.

7.2.3 Issues with the ingesting gaia-catwise data

As mentioned above we needed to remove duplicates from the cross-match results. There are also some input sources missing from the results, that is they do not appear in the matched or unmatched results. Both these issues are being investigated by the work-package team.

The gaia-catwise cross-matched data was our first large-scale ingestion into this version of Qserv (2023.5.1-rc1). We initially encountered timeout issues and one of the Kubernetes pods being evicted due to out-of-memory. These were fixed in collaboration with the Qserv developers, via an update to `qserv-ingest` and a re-configuration of one of the Qserv nodes. We later got errors when building some of the director table indices this was also solved by increasing a timeout.

Some of the issues above potentially point to an underlying problem with disk I/O speeds and we have also seen odd variations in some query times.

8 Next steps

8.1 Near infrared data fusion

In terms of the butler work we are currently investigating generating a skymap dataset and Hierarchical Progressive Survey (HiPS) image from the deep coadds. We have a use case that requires the generation of thousands of image cut-outs and the skymap is needed to drive this.

Larger sky area pipeline runs are also available for transfer and ingestion. Around 50TB of coadd data will be transferred and ingested to the UK-DAC and around 450TB of pipeline output will be transferred to Echo storage at RAL.

8.2 Cross-match

The priority is to understand how to join the cross-match results with the original survey datasets within Qserv, rather than creating and ingesting pre-joined data.

Qserv performance is a concern, and we will investigate where the bottlenecks are located and test how other storage options perform (e.g., solid state drives).

9 Conclusion

We have summarized the system for transferring, ingesting, and publishing user-generated products to the UK-DAC and making them accessible to users via the RSP. Two uses cases are described for successfully handling the data products output by two work packages.

10 References

- [1] WP3-5 DAC -DEV interface definition,
https://lsst-uk.atlassian.net/wiki/download/attachments/2862448641/WP3-5_DAC-DEV_interface_definition.pdf?api=v2, accessed 28th June 2023
- [2] WP3-11 DAC -DEV interface definition,
https://lsst-uk.atlassian.net/wiki/download/attachments/2862448641/WP3-11_DAC-DEV_interface_definition.pdf?api=v2, accessed 28th June 2023
- [3] The VISTA VIRCAM and Vera C. Rubin Observatory Joint Photometry Pipeline,
<https://www.overleaf.com/project/5f049ad6f5fc550001ab9752>, accessed 28th June 2023
- [4] Report on the Preparation for Full-Scale DAC Matches,
https://lsst-uk.atlassian.net/wiki/download/attachments/1146928/LUSC-B-37-D3.11.4_Full_Scale_DAC_Matches_report.pdf?api=v2, accessed 28th June 2023
- [5] Preliminary cross-match list,
https://lsst-uk.atlassian.net/wiki/download/attachments/766574618/preliminary_match_list.txt?version=6&modificationDate=1668087400778&cacheVersion=1&api=v2, accessed 28th June 2023
- [6] Data Butler Product Definition,
<https://confluence.lsstcorp.org/display/DM/Data+Butler+Product+Definition>, accessed 28th June 2023
- [7] D. L. Wang et al, Qserv: A distributed shared-nothing database for the LSST catalog, 2011,
https://www.researchgate.net/publication/254003056_Qserv_A_distributed_shared-nothing_database_for_the_LSST_catalog, accessed 28th June 2023
- [8] sph-partion,
<https://github.com/lsst/qserv/tree/main/src/partition>, , accessed 28th June 2023
- [9] qserv-ingest,
<https://github.com/lsst-dm/qserv-ingest/> accessed 28th June 2023
- [10] WP3.5 - Notes on ingesting SXDS into Qserv,
<https://lsst-uk.atlassian.net/wiki/spaces/LUSC/pages/3008299009/WP3.5+-Notes+on+ingesting+SXDS+into+Qserv>, accessed 28th June 2023
- [11] Felis
<https://felis.lsst.io/>, accessed 28th June 2023
- [12] Gaia and CatWISE V0.2 Cross-Match Catalogue API and Flag Filters
https://lsst-uk.atlassian.net/wiki/download/attachments/763166721/gaia_catwise_api_V1.pdf?api=v2, accessed 4^h July 2023