Rubin Multi-site Data Movement for DRP Processing

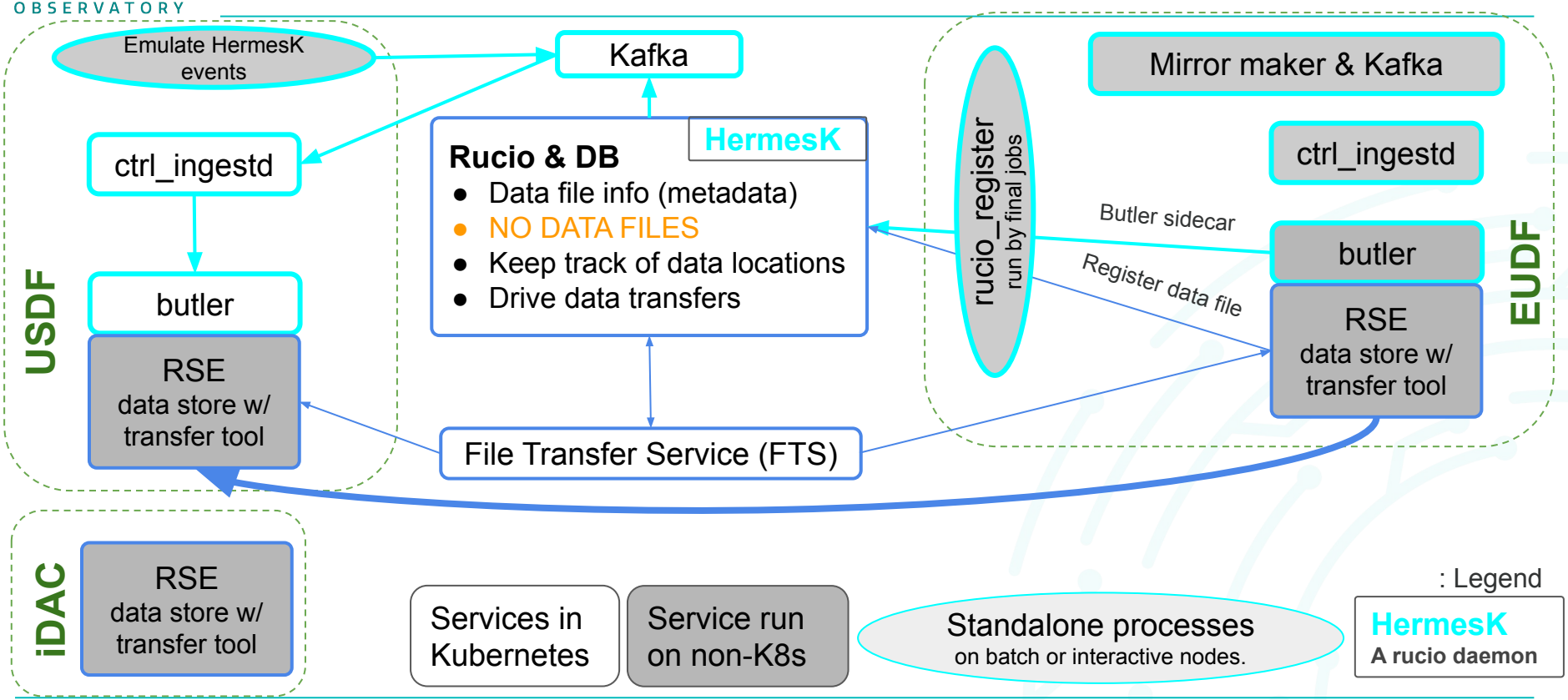
Wei Yang @ SLAC

Multi-site Data Movement

For Rubin multi-site processing and data movement, we glue LSST Butler and HEP Rucio together.

- LSST Butler DB keeps trace of astronomical datasets (usually files) though “dimension” records
  - Observation time, instrument specs, sky coordinates, dataset type, tract, patch, etc.
  - Keep trace of dataset (file) locations.
  - Organize data in collections.
  - Butler provides an API: allowing quick identifying of the relevant datasets based on search criteria.
- Rucio keeps trace data at different locations, and drive data movement
  - Doesn’t know any physics parameters (thought there are metadata for labeling)
  - Organize data in datasets and containers (equivalent to butler collections)
  - Rucio runs a set of daemons to perform data movement actions, and UI/API to interact with users
- One thing to note:
  - A Butler collection usually contains x100+ more files than a Rucio dataset.

Design of Butler + Rucio



Using the system & learning the bottlenecks

We have setup an automated system to move data. The end goal is “butler-to-butler”.

- Distribution of RAW image data is a daily activity
  - All CCD images from a focal plane are zipped together, into a 3-4 GB file size.
  - Slow pace compare to DRP
- DRP has O(10^9) small files. This is the real challenge.

We are testing / learning / improving

1. How quickly can we register data to Rucio along with butler dimension records
2. How quickly can we move the data between sites
3. How quickly (and safely) can we ingest dimension records to the destination butler.
4. Tools for verification

Benchmark of performance:

- GB/s is meaningless. **Files/hour is the benchmark.**
- Many of the performance tests used DP1 data (**1.26m** files) and DP1 butler, in a few cases, DP2 butler (1.1b files, including intermidates)

Getting data into Rucio

Identified several bottleneck and mitigations

1. Chopping a Butler repo to suitable size that fits into many rucio datasets
    - Algorithms developed for DP1 doesn't work for DP2. The latter is not just much large
        - One user particular data layout in DP2 made it very hard to develop an efficient algorithm
    - It is possible to identify butler intermediates that don't need to go to Rucio.
2. Registering to Rucio
    - Computing checksum is a bottleneck.
        - FRDF and UK data stores adler32 checksum at file creation time.
        - Not at USDF.
        - For test with DP1, we pre-calculate adler32 before this step.
    - Adding file DIDs and register each file DID to a rucio dataset took 6 hours (or ~200k file/h)
    - Adding butler dimension records to Rucio took another 100 minutes.
    - The above two are sequential operations. Can potentially parallarize the operation at file level.

Moving files between sites

SLAC to IN2P3 10h rate →

Source	Destination	VO	Submitte	Activ	Staging	S.Acti	Archiv	Finish	Faile	Cance	Rate (last 1h)	Thr.
+ davs://sdfdtn005.slac.stanfo	davs://ccdavrubin.in2p3.fr	lsst	117547	243	–	–	–	261309	3	–	100.00 %	79.89 MiB/s

IN2P3 to RAL & SLAC 1h rate →

Source	Destination	VO	Submitte	Activ	Staging	S.Acti	Archiv	Finish	Faile	Cance	Rate (last 1h)	Thr.
+ davs://ccdavrubin.in2p3.fr	davs://sdfdtn005.slac.stanfo	lsst	294097	163	–	–	–	40917	–	–	100.00 %	0.08 MiB/s
+ davs://ccdavrubin.in2p3.fr	davs://xrootd.echo.stfc.ac.u	lsst	291106	166	–	–	–	45182	–	–	100.00 %	0.04 MiB/s

IN2P3 to SLAC 1h rate →

Source	Destination	VO	Submitte	Activ	Staging	S.Activ	Archivi	Finishe	Faile	Cancel	Rate (last 1h)	Thr.
+ davs://sdfdtn005.slac.stanford.	davs://xgate.hec.lancs.ac.uk	lsst	223	3	–	–	–	–	–	–	0.00 %	–
+ davs://ccdavrubin.in2p3.fr	davs://sdfdtn005.slac.stanford.	lsst	–	62	–	–	–	102596	1920	–	98.16 %	0.11 MiB/s

More test are going on to understand each DF's storage performance limitation.

Ingest to destination butler

Ingestion to destination butler could not keep with the transfer speed at high rate

- Weird DB behavior at ~100 kfile/h.
- Likely because the DB can’t keep up ← this DB is a testing setup.
- When we limited the rate to ~60 to 75 kfile/hour, ingestion did not fail.

- ingestion doesn’t have flow control
- Rucio doesn’t get feedback from the failure of ingestion
- We are looking for mitigations or re-design
  - Mitigation: understand each DF’s DB performance limitation.

We also are looking for tools to verify ingestion (and tool to verify register to rucio)

Estimation of DR1: data to be moved

Courtesy of Brian Yanny

The division between intermediates (to be threw away) and to be transferred/saved can change depending on requirements.

DF	Fraction	Intermediate size(50P)	Transfer count/size	Move Files/Bytes/d
UK RAL	0.15	7.5PB	14M/250TB	
UK LANCS	0.10	5PB	9M/150TB	90K,1.5TB
FR IN2P3	0.40	20PB	36M/600TB	360K,6TB
US SLAC	0.35	18PB	32M/—-	Receive 500K,10TB

Summary

IF the estimated number of DR1 data rate is close to reality, the transfer system will be able to keep up.

- It requires only 20 - 25 K file/h ← Of course there is a big IF here.
- On the other hand, we can handle x3 more (before we need DB tuning)

We are gluing Butler and Rucio together.

- There is less feedback and overall flow control between them.
- Storage performance is also critical, though Rucio will ultimately takes care of that.

Understanding of the limitations of butler DB performance is critical to mitigate potential issue.

- We are also build safe guardrails around it.

DP2 has 1.1b files (including intermediates). We will continue to use it to improve our systems.

Backup Slides

Status of DP2 Processing

Processing Completed

- DP2 raw images (2025-04-15 to 2026-01-07)

Estimation of DR1: total retained data

Courtesy of Brian Yanny

datasetType	DR1 count	Single File Size	Total Size	Days
Visit_image +background	37M+37M	30MB+1MB	1100 TB	
deep_coadd+template_c	8M+8M	25MB	350 TB	50d(stage3)
object_scarlet_models	1.2M	70MB	90 TB	50d(stage3)
Totals	90M	1M files/day	1500 TB	100d

Estimation of DR1: intermediates

Courtesy of Brian Yanny

Local to DFs. Removed after DR1

datasetType	DR1 count	Single File Size	Total Size	Days
warps	300M	70MB	20PB	
Stage4a int	120M	100MB	15PB	50d(stage4)
preliminary_visit_image	35M	30MB	1PB	200d(stage1-4)
Totals	500M		50PB	200d