

Computing news

Credits:

- Quentin le Boulc'h - CC-IN2P3
- Bernard Chambon - CC-IN2P3
- Sabine Elles - LAPP
- Fabio Hernandez - CC-IN2P3
- Fabrice Jammes - LPC
- Gabriele Mainetti - CC-IN2P3

Data Preview 0

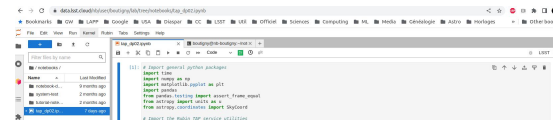
First large scale exercise to test Data Release Processing from end to end

- DP0.1 was "just" a copy of DESC DC2 in Qserv / RSP
- DP0.2 is a full reprocessing of DESC DC2
 - add difference Imaging (DIA) in WDF
 - doesn't include the DC2 DDF
 - impossible to process with the current DM stack
 - requires cell-based Coadds

DP0.2 ran on the Interim Data Facility (IDF) on Google cloud

- No job distribution on the 2 non-US DF: UKDF and FrDF
- but replication of the full exercise at CC-IN2P3 ← **Thanks Quentin !**

DP0.2 data



Caveat

- The DP0.2 butler at CC-IN2P3 is the result of the French local DP0.2 processing
- The catalogs loaded into Qserv and accessible from the notebook platform or the RSP@CC have been copied from the IDF
 - The French produced catalogs will be loaded later for comparison purpose

At CC-IN2P3:

- butler: `/sps/lst/dataproducts/rubin/previews/dp0.2/butler.yaml` \leftarrow **Need postgresQL credentials**
 - from <https://notebook.cc.in2p3.fr>
 - or directly from Slurm
- catalogs
 - SQL direct queries on Qserv
 - ADQL queries through TAP on the RSP@CC: <https://data-dev.lsst.eu> \leftarrow **Thanks Gabriele**

DP0.2 processing at CC-IN2P3

Fully documented and coordinated with DM

github.com/lstt-dm/dp02-processing/tree/FRDF/full/production

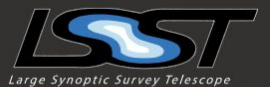
Bookmarks GW LAPP Google USA Diaspar CC LSST Util Official Sciences

lstt-dm / dp02-processing Public

<> Code Issues Pull requests Actions Projects Security Insights

rtm-029.lstt.io

Bookmarks GW LAPP Google USA Diaspar CC LSST Util Official Sciences Computing ML



RTN-029: Procedure for creating a butler repository at FrDF for Data Preview 0.2

Fabio Hernandez, Quentin Le Boulch, Jim Bosch, Hsin-Fang Chiang, James Chiang, Tim Jenness and Brandon White

Latest Revision: 2022-03-31

1 Introduction

In this note we document the required input datasets and the procedure we followed at the Rubin French Data Facility (FrDF) for creating and populating a butler repository for the needs of image processing for preparing Data Preview 0.2 [1].

1 Introduction
2 Input Datasets

FRDF dp02-processing / full / production /

This branch is 67 commits ahead, 116 commits behind main.

QLeB Add step7

- step1 Update requestMemory.yaml to use hor
- step1_rescue Add template file for step2
- step1_rescue2 Add step1_rescue2
- step2 Create new output collection for step2
- step2_rescue Add step2_rescue

jira.lsttcorp.org/browse/PREOPS-1145

Bookmarks GW LAPP Google USA Diaspar CC LSST Util Official Sciences Computing ML Media Généalogie Astro Horloges Other bookmark

Jira Software Dashboards Projects Issues Boards Calendar Tests Pivot Reports Scrum Standup Create Search

Pre-Operations / PREOPS-1145

DP02 production processing at FRDF

Edit Comment Assign More To Do Ask For Review Done

Email Pivot Report Export

Details

Type: Story Status: **IN PROGRESS** (View Workflow)
Priority: ★ Undefined Resolution: Unresolved
Component/s: Data Production
Labels: FrDF

Description

This is the main issue to track the DP0.2 campaigns at FRDF.

People

Assignee: Quentin Le Boulch
Reporter: Quentin Le Boulch
Watches: Dominique Boutigny, Fabio Hernandez, Peter Love, Quentin Le Boulch, Wil O'Mullane, Yusra AISayyad
Votes: Vote for this issue
Watches: Stop watching this issue

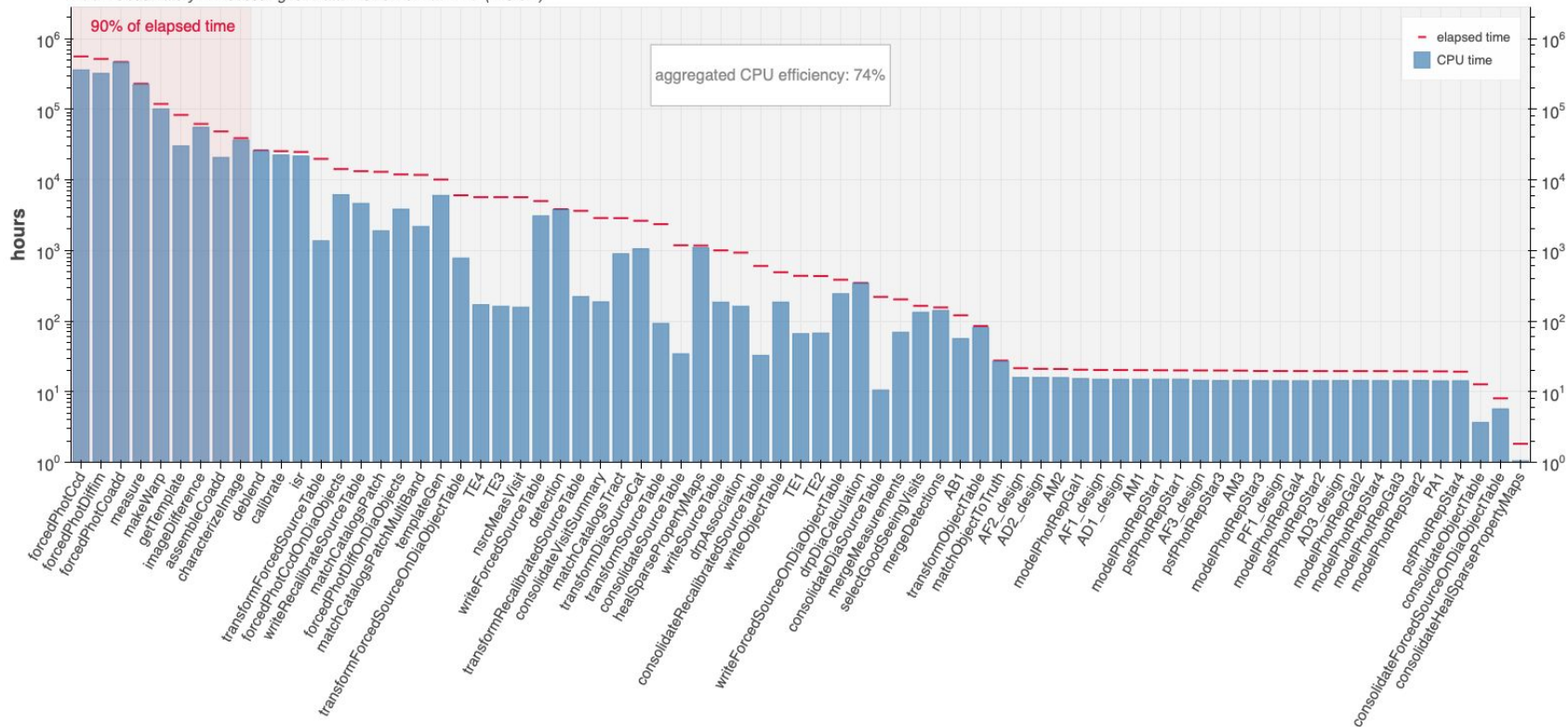
DP0.2 @CC - a few numbers

- From April to October '22
- 7 steps to complete the production
- up to 3000 simultaneous Slurm jobs
- 57,903,740 quanta (from quantum graph)
 - 2,347,306 elapsed hours
- Butler repository details:
 - input dataset : 51 TB - 2.9 M files
 - products: 3 PB - 54 M directories - 201 M files (including intermediates)
 - registry database (postgreSQL) : 314 GB

Running DP0.2 at CC-IN2P3 allowed to collect crucial metrics

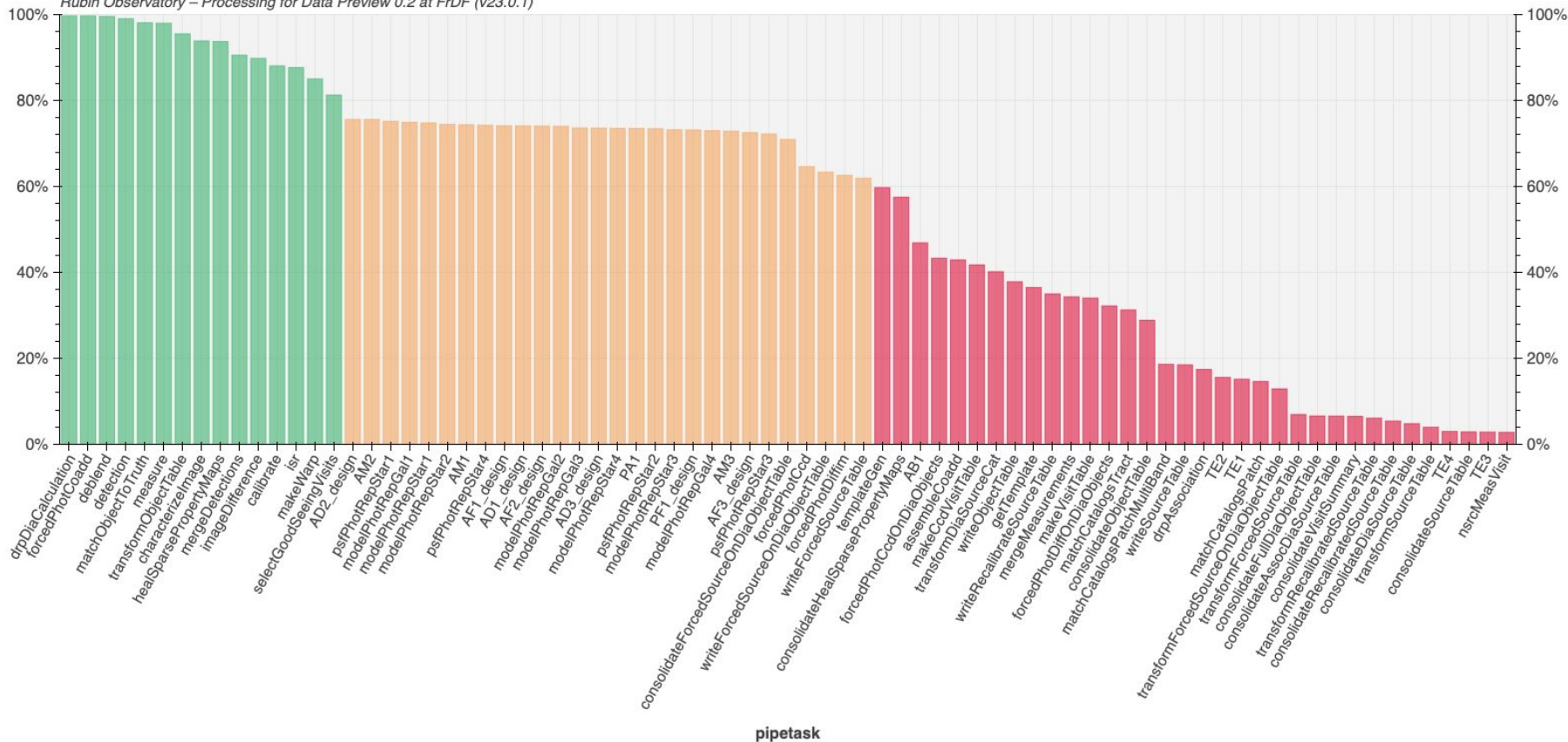
Elapsed and CPU time spent by pipetask kind

Rubin Observatory – Processing for Data Preview 0.2 at FrDF (v23.0.1)



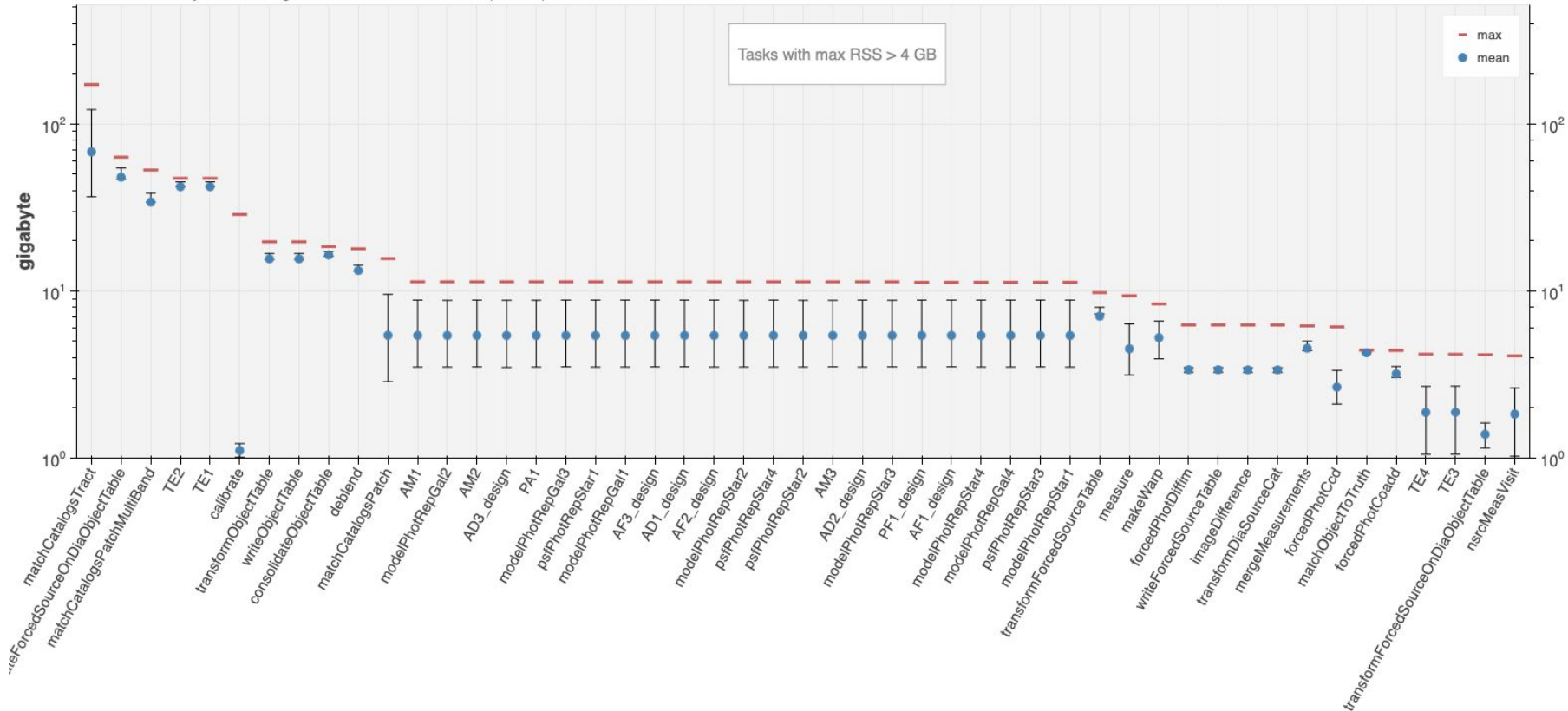
CPU efficiency aggregated by pipetask kind

Rubin Observatory – Processing for Data Preview 0.2 at FrDF (v23.0.1)



Memory consumption by pipetask kind

Rubin Observatory – Processing for Data Preview 0.2 at FrDF (v23.0.1)



Only pipetasks with peak memory >4 GB are shown

Interval 30m metadata pool 6 Instance cccephadm20.in2p3.fr:9283 data pool 4

More raw numbers about SPS LSST@CCIN2P3 OVERVIEW

Data Pool Throughput



	Mean	Last *	Max	Min
Read Bytes	0 B/s	0 B/s	0 B/s	0 B/s
Read Bytes	9.50 GiB/s	0 B/s	26.6 GiB/s	0 B/s
Read Bytes	0 B/s	0 B/s	0 B/s	0 B/s
Write Bytes	0 B/s	0 B/s	0 B/s	0 B/s
Write Bytes	980 MiB/s	0 B/s	2.89 GiB/s	0 B/s
Write Bytes	0 B/s	0 B/s	0 B/s	0 B/s

Issues - lessons learnt

Several issues encountered - some are site specific (Google cloud \neq Slurm + CephFS)

A few examples:

- ❖ Registry database load
 - use a local execution butler
- ❖ Quantum graph generation memory usage
 - submission from 500 GB RAM
- ❖ Parsl instability with too many tasks (~500k tasks)
 - use task clustering
- ❖ Performance issues with some tasks heavily impacting CephFS
 - Patch butler to copy some files on the local disk
- ❖ Silent file corruption when a job is killed
- ❖ ...

Next steps with DP0 dataset

- Need to generate HIPS map for visualization purpose but requires a risky migration of the butler registry database schema
- Need to redo part or all the DP0 processing using dCache through webdav instead of CephFS / POSIX
 - webdav - butler interface written by Bastien Gounon and rewritten and maintained by Fabio
- If it works as expected, this will have a significant impact on the storage cost

A word on the RSP

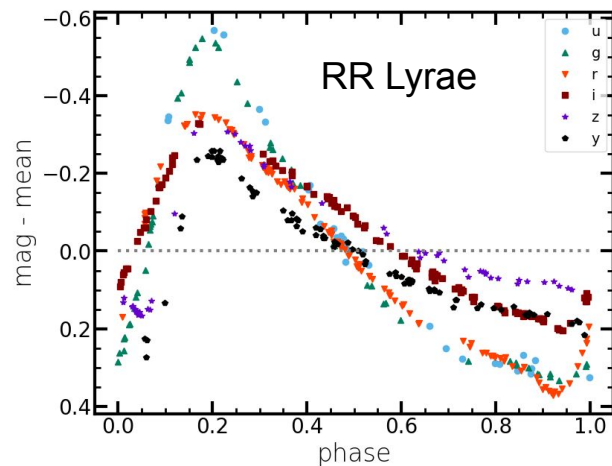
The RSP on Google cloud is fully operational and is working well <https://data.lsst.cloud/>

- If you are a delegate try it and report your experience
 - <https://community.lsst.org/c/support/dp0/49>
- If you are not, you can ask to become one to Melissa Graham

There are a lot of interesting documentation / resources / tutorial in:

- <https://dp0-2.lsst.io/tutorials-examples/index.html#>
- <https://github.com/rubin-dp0/tutorial-notebooks>

The examples with DIA objects are new and interesting



Qserv

- We have a very powerful Qserv cluster deployed at CC-IN2P3
 - significantly faster than the one at IDF
 - sample full scan query at CC (direct SQL query) : ~2 minutes
 - same query at IDF (through TAP): ~15 minutes
- Fabrice developed a Qserv ingest tool integrated to kubernetes
 - Efficiently and reproducibly ingestion of large catalogs
 - Need to create a huge temporary set of csv files (text format)
 - because MariaDB is optimized for csv
- Sabine and Fabrice working to improve the ingestion tool to avoid the csv step
 - direct ingestion from the *parquet* files (still requires some duplication but probably needs less intermediate storage)

Data Previews and Data Releases Schedule

From [RTN-011](#)

Data Preview/Release			FY23	2023	FY24	2024	FY25	2025	FY26	2026	FY27	2027	FY28	2028
DP0.1	DC2 Simulated Sky Survey	June 2021												
DP0.2	Reprocessed DC2 Survey	June 2022												
DP1	ComCam On-Sky Data	Mar 2024 - Jul 2024				X								
DP2	LSSTCam On-Sky Data	Jan 2025 - Aug 2025												
DR1	LSST First 6 Months Data	Oct 2025 - May 2026												
DR2	LSST Year 1 Data	Oct 2026 - May 2027												
DR3	LSST Year 2 Data	Oct 2027 - May 2028												

Not realistic anymore - I guess that we will have a Data Preview / Release before August 2025

In any case we need at least 1 large scale test with job distribution on the 3 production sites

- HSC PDR3 considered for this exercise

Budget

Detailed cost model established by Fabio

- See [ATRIUM-730702](#)
- Based on input assumptions provided by DM

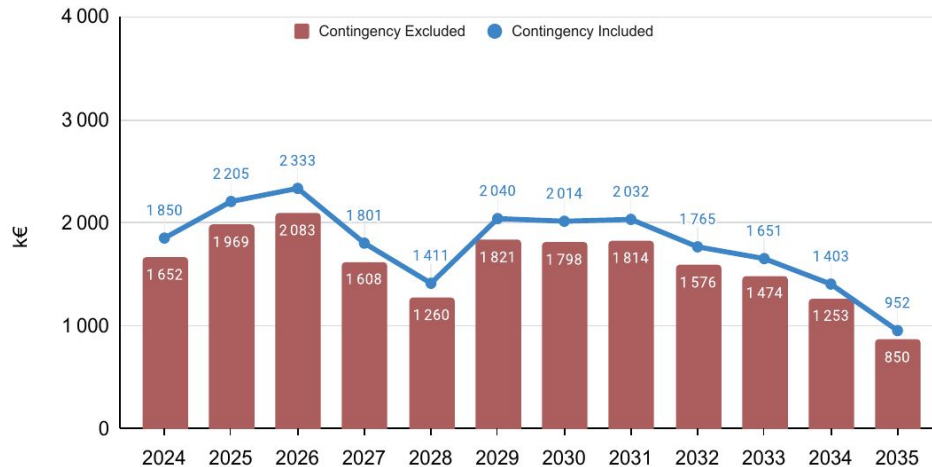
Study various hypothesis

- See [ATRIUM-753740](#)
- Identify possible cost reduction
 - Improve efficiency
- Consider cost stretching (increase processing time)

Presented to Vincent Poireau and CC director

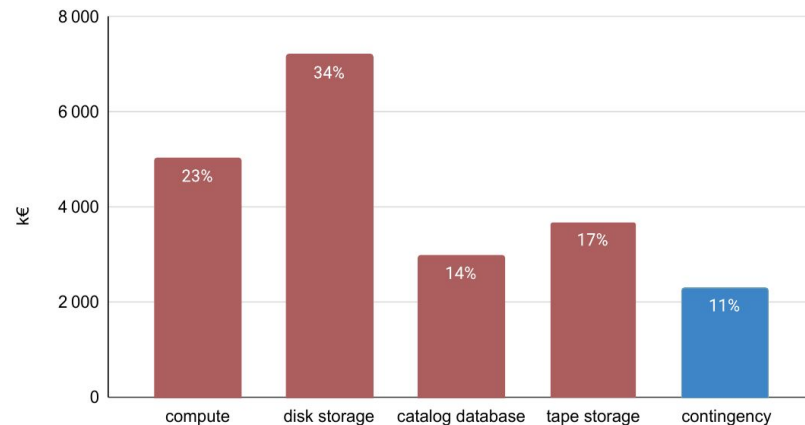
No clear way to significantly reduce the cost without impacting our scientific return

EQUIPMENT BUDGET PROFILE

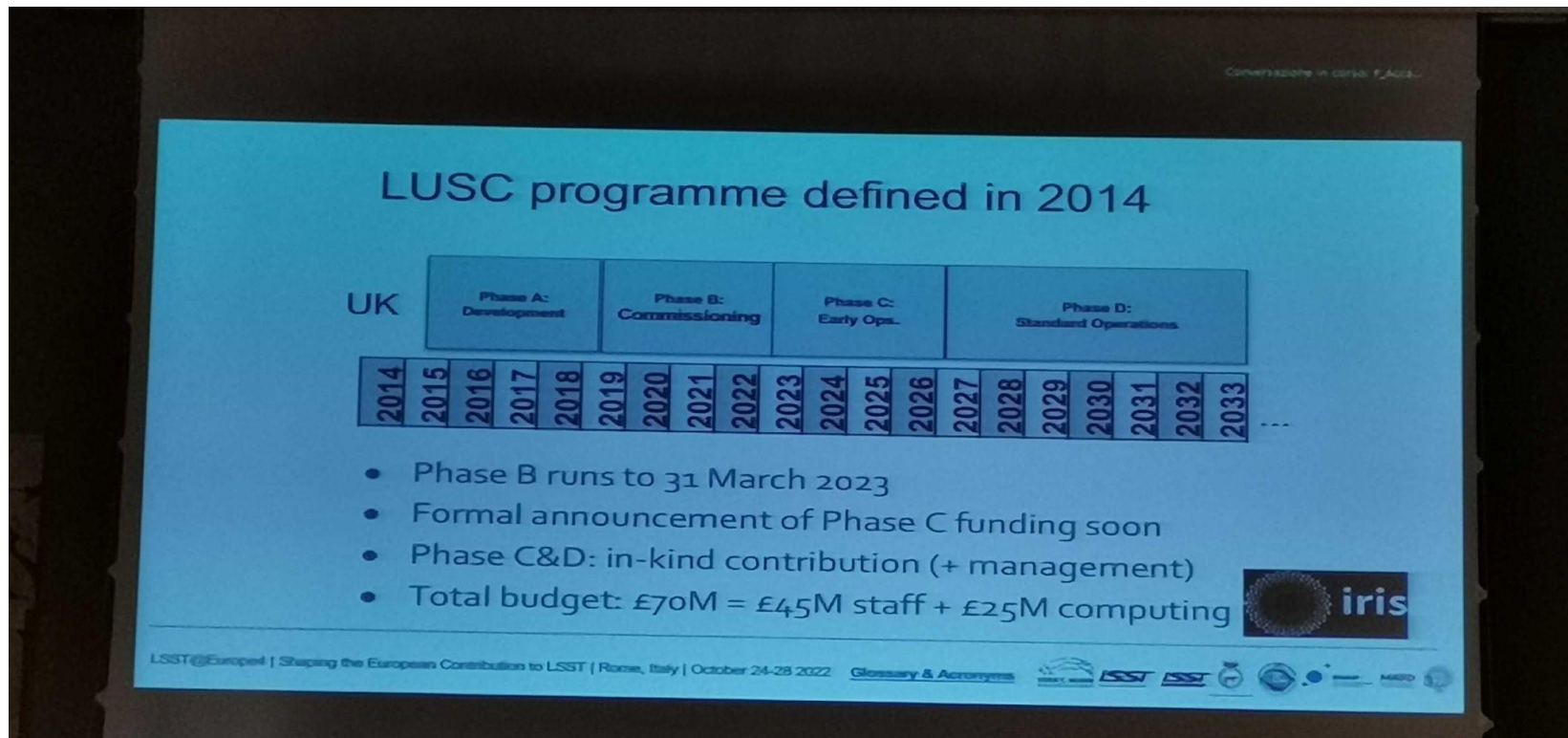


EQUIPMENT BUDGET BREAKDOWN

(2024-2035)



Pour comparaison... UKDF presented at LSST@Europe4



Documentation

<https://doc.lsst.eu/>

The screenshot shows the LSST-France User Guide website. The page title is "LSST-France User Guide" and it includes a search bar and a navigation menu. The main content area is titled "LSST-France User Guide" and contains a welcome message, a note, and sections for "GETTING STARTED" and "COMPUTING ENVIRONMENT".

LSST-France User Guide

1.0

Search docs

GETTING STARTED

- Collaboration tools

COMPUTING ENVIRONMENT

- Working Environment at CC-IN2P3
- Login Farm
- Batch Farm
- Data Storage and File Systems
- Software
- Monitoring and Dashboards

HOW TO

- Overview
- How to customize your SSH client
- How to decide where to store my data
- How to share data with your collaborators
- How to use the LSST science pipelines
- How to activate the DESC software environment
- How to execute LSST-enabled JupyterLab notebooks
- How to use stacker to run JupyterLab notebooks
- How to create your own data butler

PROVIDING FEEDBACK

- Provide Feedback

Docs » LSST-France User Guide

LSST-France User Guide

Welcome to the *LSST-France User Guide*. Here you will find supplemental information about the Rubin Observatory Legacy Survey of Space and Time (LSST) specific to the LSST community in France.

Note

This space is a permanent work in progress. Please see [Providing Feedback](#) for details on how you can help improving it.

GETTING STARTED

- Collaboration tools
 - Project-wide tools
 - LSST-France tools

COMPUTING ENVIRONMENT

- Working Environment at CC-IN2P3
 - Overview
 - How to Get Help
 - Account Setup
 - Operations Status
 - Scheduled maintenance
- Login Farm
- Batch Farm
- Data Storage and File Systems
 - Home directory: `$HOME`
 - Shared group area: `/rps/throng/lsst`
 - Shared group area (large datasets): `/rps/lsst`
 - Interactive working area: `/scratch`
 - Batch job working area: `$TMPDIR`
 - Archival storage

Slack channels

The screenshot shows a list of Slack channels. The channels are listed in a dark blue background with white text. The channels are: # in2p3, # in2p3-auxteldm, # in2p3-commissioning, # in2p3-cvmfs, # in2p3-dask, # in2p3-desc-dc2, # in2p3-dp0-private, # in2p3-focal-plane, # in2p3-gen3, # in2p3-lapp, # in2p3-mediation-edi, # in2p3-ml, # in2p3-notebook, # in2p3-ops, # in2p3-qserv, and # in2p3-workflow.

- # in2p3
- # in2p3-auxteldm
- # in2p3-commissioning
- # in2p3-cvmfs
- # in2p3-dask
- # in2p3-desc-dc2
- # in2p3-dp0-private
- # in2p3-focal-plane
- # in2p3-gen3
- # in2p3-lapp
- # in2p3-mediation-edi
- # in2p3-ml
- # in2p3-notebook
- # in2p3-ops
- # in2p3-qserv
- # in2p3-workflow

Don't hesitate to ask questions on the relevant channel (especially #in2p3), you will likely get a prompt answer

Tools

Notebook platform : <https://notebook.cc.in2p3.fr/>

- Up to 24 GB of memory
- Allows to access the Rubin and DESC datasets
 - butler
 - GCRCatalogs
 - Qserv queries

Dask :

- A powerful tool to parallelize the execution of computing tasks
- Natively interfaced with pandas dataframes
- Now fully integrated with the notebook platform
 - dask4in2p3 package developed by Bernard Chambon
 - See demo / tutorial tomorrow

The screenshot displays a Dask notebook environment. On the left, a sidebar lists various tool categories such as BANDWIDTH TYPES, CLUSTER MAP, CPU, GPU MEMORY, and WORKERS. The main area contains a code editor with Python code for data processing using Dask and Pandas. The code includes operations like grouping data by halo ID, calculating richness, and merging datasets. On the right, there are monitoring dashboards: 'CPU Utilization' showing a bar chart, 'Workers Memory' showing a heatmap, and 'Task Stream' showing a vertical bar chart of task execution over time.

Under beta-test with some expert users at the moment

- But ask if you want to be added to the testers
- A lot of tests done by Mickael
- A few limitations that are investigated by Bernard

Plan to open to everybody in January

USDF

The USDF is hosted at SLAC

- RSP@USDF: <https://usdf-rsp.slac.stanford.edu/>
- S3DF is the interactive cluster + Slurm batch cluster

- See documentation from Yousuke Utsumi:
<https://confluence.slac.stanford.edu/pages/viewpage.action?spaceKey=~youtsumi&title=Using+USDF+on+S3DF>
- and also: <https://s3df.slac.stanford.edu/public/doc/#/>

SLAC on-boarding procedure: <https://developer.lsst.io/usdf/onboarding.html>

Example of a raw auxTel image (with hologram) displayed on the RSP@USDF

