The Rubin Observatory’s Legacy Survey of Space and Time DP0.2 processing campaign at CC-IN2P3

Fabio Hernandez, Quentin Le Boulc’h, Gabriele Mainetti

CHEP 2023, Norfolk, USA, 05.11.2023
• Legacy Survey of Space and Time
• Rubin data processing pipeline
• Data Preview 0.2 (DP0.2)
• Workflows of pipeline tasks
• Execution of the processing campaign
• Perspectives
Legacy Survey of Space and Time (LSST), at the Vera Rubin Observatory:

- Science themes: dark energy, dark matter, but also solar system and transient objects
- 8.4m telescope at Cerro Pachon (Chile)
- 3.2 Gpix camera
- All visible sky (~18000 deg2) in 6 bands
- 10 years survey starting from 2025
Legacy Survey of Space and Time

LSST will produce:

• Up to 10 million alerts per night (transient and variable sources)
• Data products (images and catalogs of 20 billion galaxies and 17 billion star)
• 20 TB per day = 60 PB of raw data aggregated over 10 years

CC-IN2P3 contributions:

• Host a full copy of raw data + selected annual products
• Perform 40% of Data Release Processing (25% UK, 35% US)
• Rubin Science Platform: web service for access, visualization and analysis of data products
• Catalog database: Qserv
Rubin data processing pipeline

- **Process raw images** and calibrate them to remove any artefact from instrument
- Merge images together ("**coaddition**") to improve signal/noise
- **Measure differences** between images to detect changes
- Detect objects, measure their properties and populate **catalogs**
Data Preview 0.2 (DP0.2)

Data Preview 0.2 exercise:
• Reprocessing of simulated data using latest pipelines
• Integration test of processing pipelines, data management system, and infrastructure
• Introduction of workflow automation

Simulated Rubin images generated by the Dark Energy Science Collaboration:
• 300 deg2 (full survey ~ 20 000 deg2)
• 5 years (full survey: 10 years)
  ➔ approximately 0.5% of the full survey
• 3 M files, 50 TB in total
Workflows of pipeline tasks

- Set of ~ 80 pipeline tasks, grouped in 7 steps
- Each step processes data at different levels: single CCD, full visit, coadded images, catalogs…
Workflows of pipeline tasks

• Pipeline tasks depend on each other through data production / consumption
• Some of the steps have rather straightforward workflows:
Workflows of pipeline tasks

- Pipeline tasks depend on each other through data production / consumption
- Others are more complex:
Workflows of pipeline tasks

• Each task must be executed between 1 and 3 millions times
• Tasks execution time goes from few seconds to 24+ hours
• Memory usage for each task varies from 1 GB to 200+ GB and depends also on input data

→ Automation of processing workflows!

Workflow automation provided by two components:
• Rubin **Batch Processing Service**: generates the workflow from description of pipeline and input data
• **Parsl**: execute the processing workflows on our computing platform
  • Library to enable Python parallelism on various computing resources
  • Can submit jobs to our local Slurm cluster
  • Provides pilot jobs through its HighThroughputExecutor feature
  • Scalable
Execution of the processing campaign

Data Preview 0.2 exercise allowed us to perform integration and scalability tests of the whole infrastructure and software environment:

- Rubin data access system and its database (PostgreSQL)
- Processing pipelines
- Workflow and job management system
- Shared filesystem (CephFS)

Resources usage:
- Used up to **3,000 simultaneous Slurm jobs**
- Executed 57,903,740 tasks which consumed 2,347,306 elapsed hours
- Generated **3 PiB** of data (**201 M files**)
- Database (metadata): 314 GiB
Some of the challenges we have faced:

• Huge workflows with very high number of tasks (~ millions)
  ➜ Define sub-workflows and gather tasks together

• Intensive access to the PostgreSQL database
  ➜ Use local sqlite databases for job access, then merge everything in the central database

• CephFS storage system performance: very good behavior overall but issues for some specific, very I/O intensive tasks
  ➜ Copy the files to the local disk of the compute node
Execution of the processing campaign

- Very heterogeneous jobs with unknown resources requirements
  - Need for measuring the task resources usage in order to tune the jobs characteristics and size the infrastructure
- Rubin pipeline execution framework provides CPU and memory usage measurements
- Metrics can then be extracted and analyzed in Jupyter Notebooks to build useful plots
Execution of the processing campaign

- Very good CPU efficiency of the most computing intensive tasks
- Good overall CPU efficiency: 74%
Perspectives

- Similar exercise using a dCache storage system ongoing
- Ongoing tests using PanDA to execute workflows
- Optimisation of pipeline CPU and memory usage
- Provide a Parsl installation for Rubin users at CC-IN2P3
Thank you for your attention!