Qserv integration into science pipelines

Astronomy ESFRI & Research Infrastructure Cluster
ASTERICS - 653477

Nicolas Chotard – Fabrice Jammes

LSST France – Paris – Mars 2017

Goals

- Test Qserv on real data processed through the LSST stack
  - Different queries (magnitudes, position, etc.)
  - Different configuration of the DB (number of stripes and chunks)
  - Different catalogs (sources, coadds)
  - ➢ Test its capabilities and performances
- Qserv integration into science analysis pipelines
  - Automatic inclusion of stack-processed data into a Qserv instance
  - Direct queries in this database from a science pipeline
  - Construction/test of python tools to query the data

➔ Test case: Clusters pipeline
  - Galaxy cluster mass estimate
  - LSST stack data used in all steps of the analysis
  - CFHT data already processed
  - 5 filters, several areas of the sky

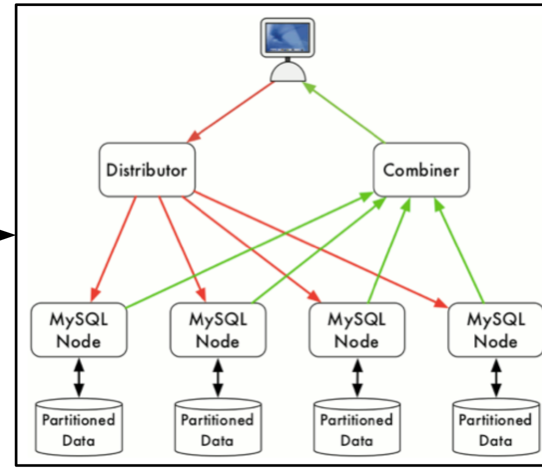
Data access

Configuration file

```
{ "ra": 340.83,
 "dec": -9.59,
 "filter": ["u", "g", "r", "i", "z"],
 "butler": "/yourpath/output/coadd_dir",
 "radius": "0.4 degree",
 "patch": ['1,1', '1,3', '1,2', '1,4', '1,5'],
 "ccd": [2, 5, 18]
 "keys":{'deepCoadd_meas': ["coord*", "id",
                                         'detect_isPrimary'],
         'deepCoadd_forced_src':["coord*",
                                               "objectId"],
         'forced_src': ["coord*", "id"]
        }
}
```

Catalogs

Qserv instance



Astropy tables in local HDF5 files

Python interface

Local queries

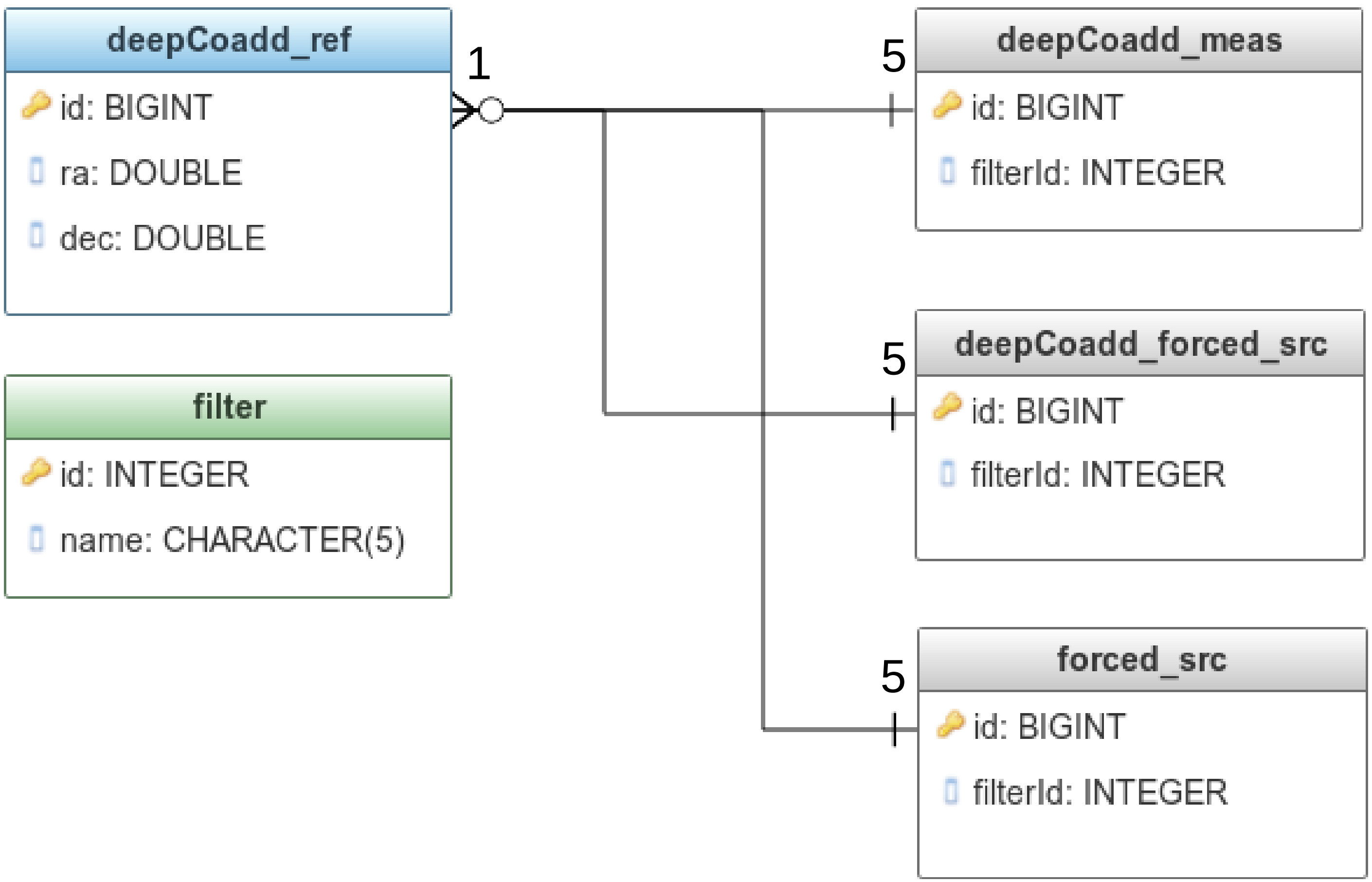
Distant queries

Clusters analysis

From butler to Qserv

- Access Qserv
  - Qserv & stack installed in Docker containers on NCSA cloud
    - 1 master, 4 workers
    - Easy and fast install (time mostly spent on downloading the containers)
    - Upgrade for latest versions of Qserv and DM stack possible at any time
  - Suitable for test only
    - Easy install, easy update, but on a development platform
    - But suitable for the tests started here
    - Long term use of Qserv in the context of analysis: CC-IN2P3
- From stack output format to Qserv input format
  - No automatic way to go from one to the other
  - Started a python script to automatize this step
  - But a few important things are still to be clarified
    - **LSST data schema**
    - **Expected input format of Qserv**

DB schema (so far)



Qserv input format

As I understand it for now

- common.cfg - database configuration
  - name, number of stripes, input format description, director table name
- description.yaml – description of the DB schema
  - File format
  - Table list (the schema)
- one_table.cfg – configuration for a given table
  - Primary key
  - Coordinate keys
  - List of all keys
  - Configuration for partition
- one_table.csv – the data in SQL format
- one_table.sql – main SQL commands to create the table

Status

- Data
  - CFHT data of cluster MACSJ2243.3-095
  - 1 filter & 2 tables
    - g filter
    - *deepCoadd_meas* & *deepCoadd_forced_src* tables
- Loading them in Qserv
  - Creation of a new test case in *qserv_testdata* github repo (used for continuous integration)
  - Produce the appropriate files for this small dataset
  - Loaded them in Mysql and Qserv DBs
  - Construct and test basic queries for these DBs
    - **identical results! → first test passed!**

Next steps

- Short term
  - Progress in understanding how Qserv works
  - Progress in describing the LSST data schema
    - What are the table of interest? Their relationship?
  - Load all data for one cluster
    - All available filter and tables
    - Text more complex queries
    - Compare results from Qserv queries and *Clusters* table filtering
- Longer term
  - Qserv @ CC-IN2P3
  - Automatic ingestion of new cluster data in the CC-IN2P3 Qserv instance
  - Python tools to query these data
  - Implementation in the *Clusters* (or other) science pipeline(s)

Summary

- Process a set of data, and produce the catalogs
- Create a Qserv instance that we can use for test
- Load catalogs in Qserv
- Create a set of queries to test the system
- Implement that into the Clusters pipeline
- Extend that to other analysis

Notes and codes can be found there

- https://github.com/nicolaschotard/qservi