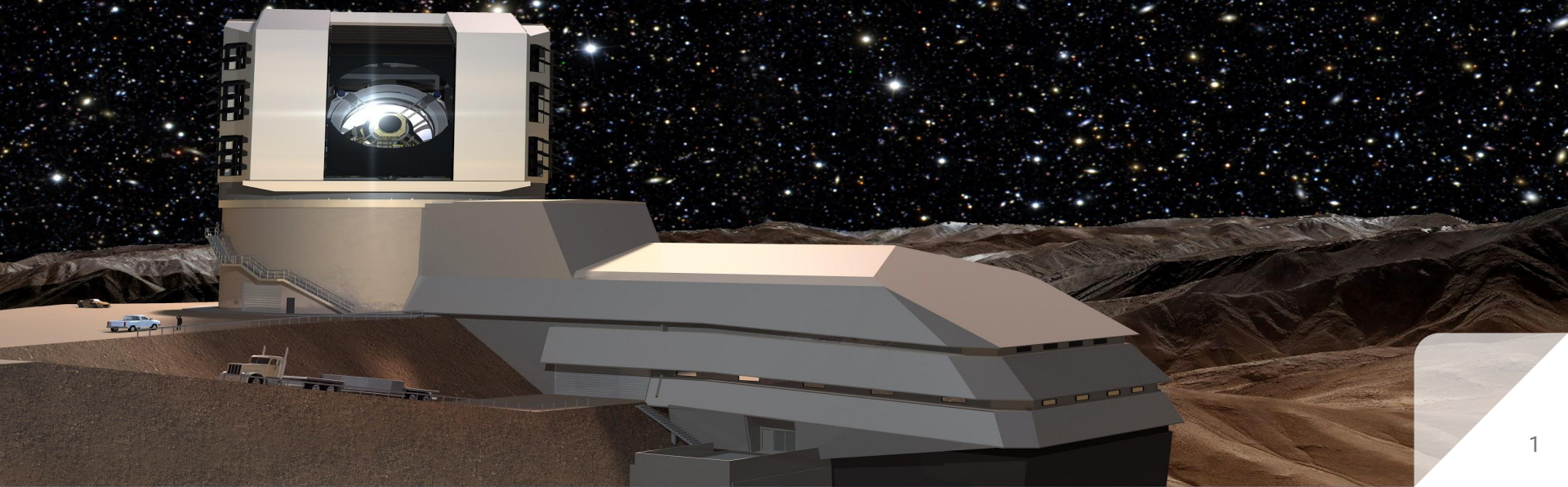
Cosmic Peta-Scale Data Analysis at IN2P3

Fabrice Jammes

Scalable Data Systems Expert
LSST-France Technical Coordinator for Databases



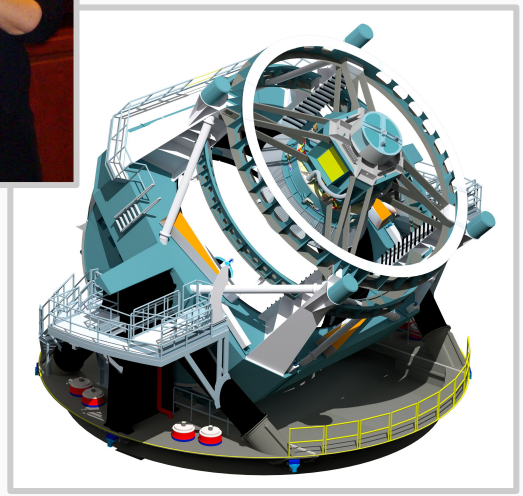
LSST in short

Large Synoptic Survey Telescope

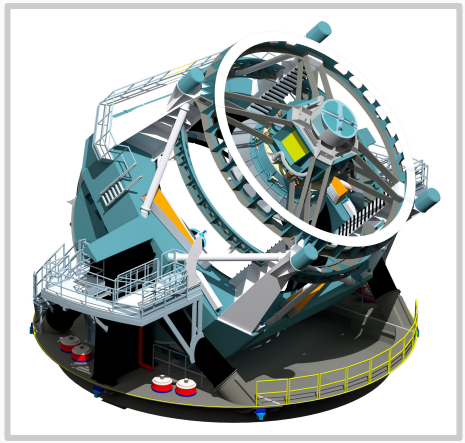
Large aperture, wide-field, ground-based survey telescope
The largest imager ever built for astronomy

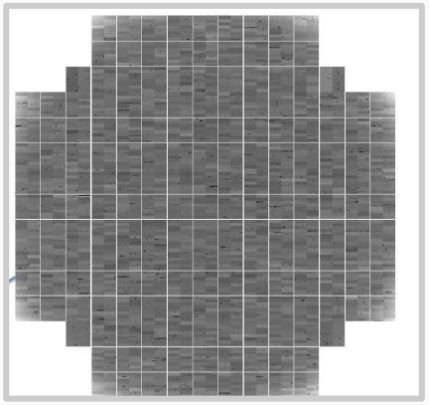
Characteristics

- ★ All visible sky in 6 bands
- ★ ~20000□
- ★ 15 seconds exposures, 1 visit/3 days
- ★ During 10 years!
- ★ **60 PB of raw data**



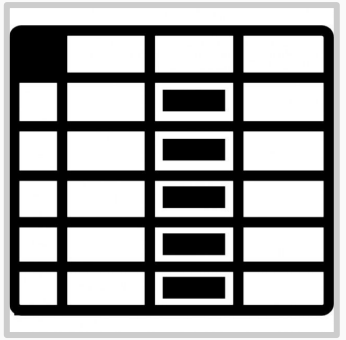
80+ PB of astronomical catalog





Raw data

Processed image



Catalog (stars, galaxies, objects, sources,transients, exposures, etc.)

LSST will build a catalog of 20 billion galaxies and 17 billion stars and their associated physical properties

Who we are

Database and Data access team

- ★ 10 engineers at SLAC + 1 LPC-IN2P3 (~10 FTE)
  - *Software development*

Operations teams

- ★ 5 engineers at CC-IN2P3
  - *Large Scale development platform*
  - *System administration, Monitoring*
- ★ 5 engineers at NCSA
  - *Prototype Data Access Center*

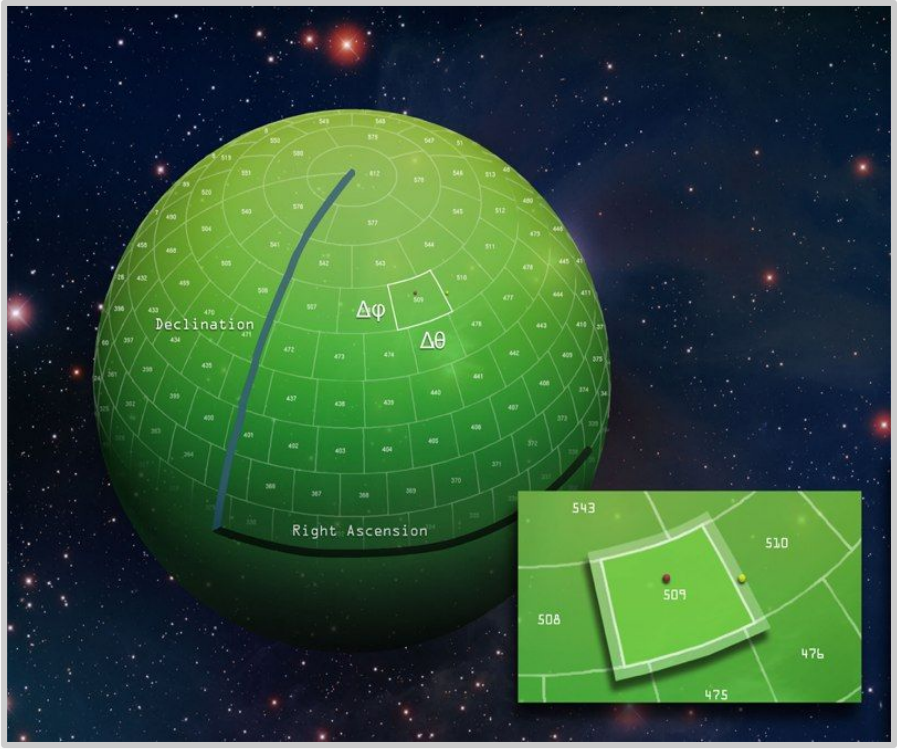
Research and development

- ★ 5 engineers at LPC-IN2P3/LIMOS (1.5 FTE)
  - *Data-loading*
  - *Cloud-computing, containers, CEPH*
  - *Large Scale Continuous integration*

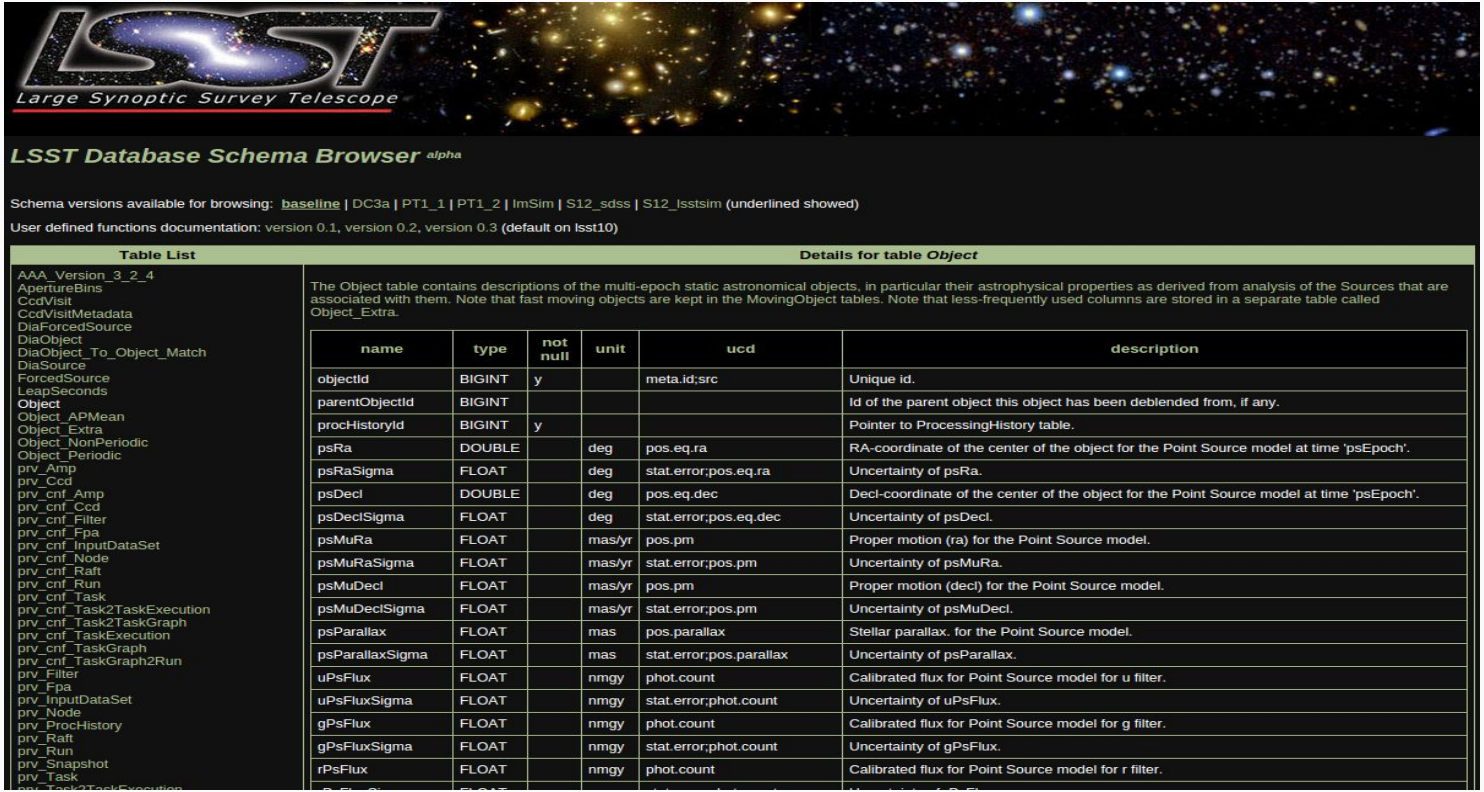
What we do

Data Access and **Database**

- ★ Data and metadata
- ★ Images and databases
- ★ Persisting and querying
- ★ For pipelines and users
- ★ Real time Alert Production and annual Data Release Production
- ★ For Archive Center and all Data Access Centers
- ★ For USA, France and international partners
- ★ Persisted and virtual data
- ★ **Estimating, designing, prototyping, building, and productizing**



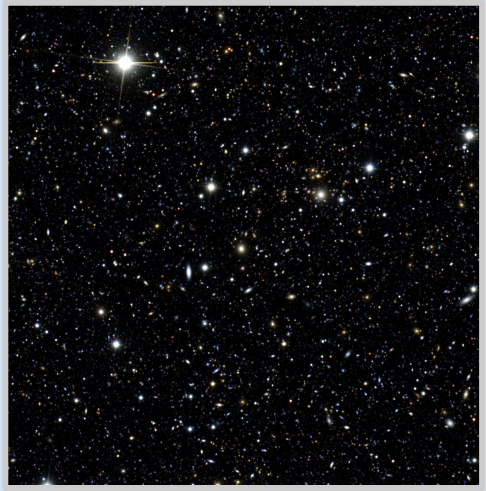
Database schema



http://ls.st/s91

Data

Images
Persisted: **~38 PB**
Temporary: **~½ EB**



★ ~3 million “visits”
★ ~47 billion“objects”
★ ~9 trillion “detections”

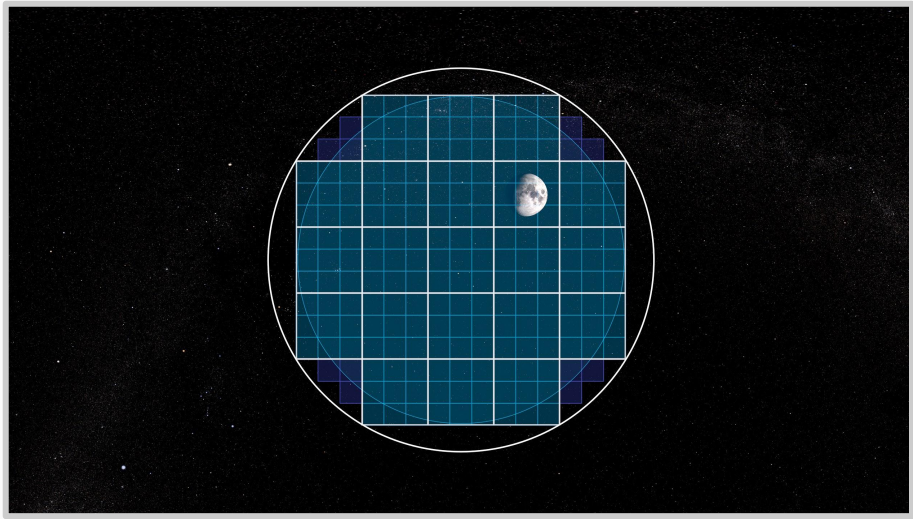
★ Largest table: **~5 PB**
★ Tallest table: ~50 trillion rows
★ Total (all data releases, compressed): **~83 PB**

Ad-hoc user-generated data
Rich provenance

Production data

Database

- ★ Real-time Alert DB.
  No-overwrite updates between Data Releases
  Real-time replica of Alert Prod DB for analytics.
  No long-running analytics here
- ★ **Immutable Database (+user workspaces)**
  Released annually. Immutable
  2 most recent releases on disk



Images

- ★ raw: 2 most recent visits for each filter
- ★ coadds and templates: for 2 most recent releases
- ★ raw calibration: most recent 30 days
- ★ science calibrated: most recent 30 days
- ★ observatory telemetry: all

Analytics

Aiming to enable majority of analytics via database
Aiming to enable rapid turnaround on exploratory queries

In a region
Get an object or data for small area - <10 sec
Across entire sky
Scan through billions of objects - ~1 hour
Deeper analysis (Object_*) - ~8 hours
Analysis of objects close to other objects
~1 hour, even if full-sky
Analýsis that requires special grouping
~1 hour, even if full sky
Time series analysis
Source, ForcedSource scans - ~12 hours
Cross match & anti-cross match with external catalogs
~1 hour

Sizing the system for
~100 interactive +
~50 complex
simultaneous DB queries.

Same for images

APIs

Metadata

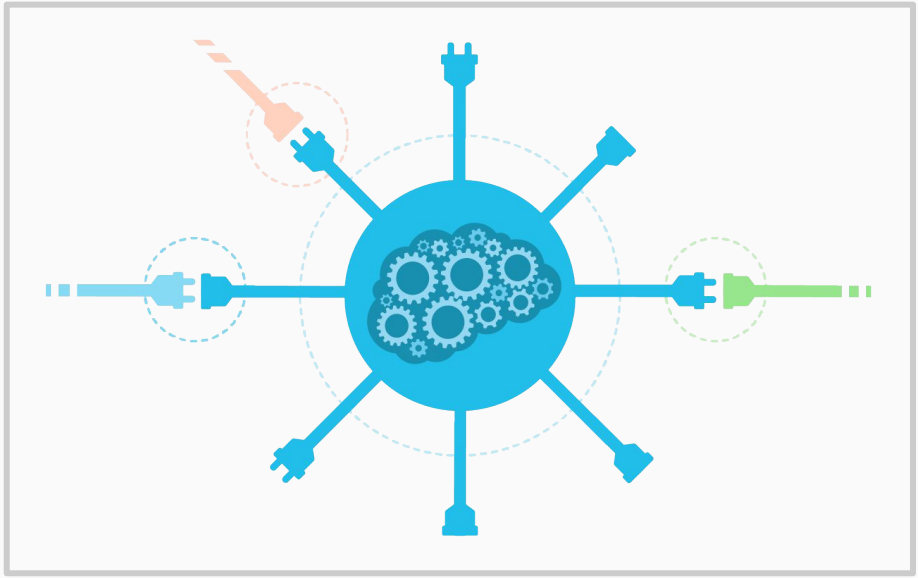
- ★ RESTful WebServ

Images

- ★ RESTful ImageServ

Databases

- ★ RESTful DbServ
- ★ SQL92 +/-, **MySQL-like DBMS**
- ★ Next-to-database python-based



Additions (SQL92 +)

Spatial constraints

- ★ qserv_areaspec_box(lonMin, latMin, lonMax, latMax)
- ★ qserv_areaspec_circle(lon, lat, radius)
- ★ qserv_areaspec_ellipse(semiMajorAxisAngle, semiMinorAxisAngle, posAngle)
- ★ qserv_areaspec_poly(v1Lon, v1Lat, v2Lon, v2Lat, …)

```
SELECT objectId
FROM   Object
WHERE qserv_areaspec_box(2,89,3,90)
```

Current restrictions (SQL92 +)

Only a SQL subset is supported

For example:

- ★ Spatial constraints (must use User Defined Functions, must appear at the beginning of WHERE, only one spatial constraint per query, arguments must be simple literals, OR not allowed after area qserv_areaspec_*)
- ★ Expressions/functions in ORDER BY clauses are not allowed
- ★ Sub-queries are NOT supported
- ★ Commands that modify tables are disallowed
- ★ MySQL-specific syntax and variables not supported
- ★ Repeated column names through * not supported

Selected Common Query Types

SELECT sth FROM Object
- massively parallel

SELECT sth FROM Object WHERE qserv_areaspec_box(....)
- selection inside chunks that cover requested area, in parallel

SELECT sth FROM Object JOIN SOURCE USING (objectId)
- massively parallel without any cross-node communication

SELECT sth FROM Object WHERE objectId = <id>
- quick selection inside one chunk

Common queries – see http://ls.st/ed4

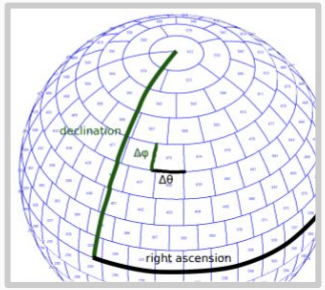
Qserv under the hood

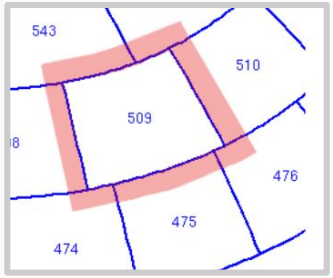
Implementation Strategy

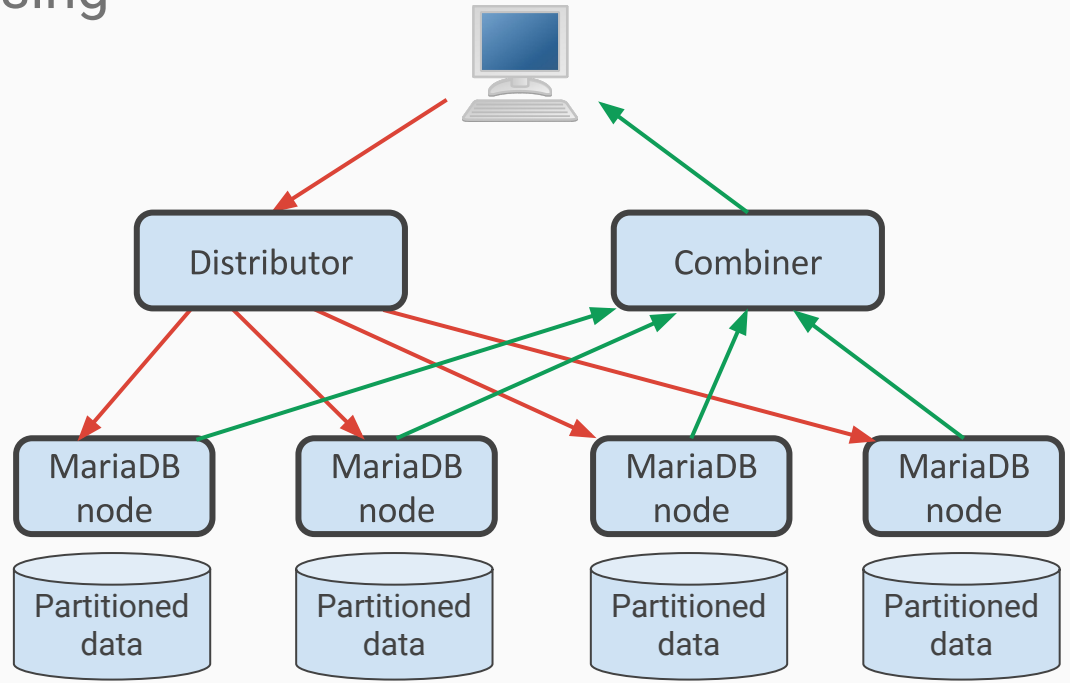
- ★ 100% Open source
- ★ Keep it flexible
- ★ Hide complexity
- ★ Reuse existing components:
  - MariaDB, MySQL Proxy, XRootD, Google protobuf, Flask
- ★ Plus custom glue
  - C++, a bit of python, some ANTLR
  - Lots of multithreading, callbacks, mutexes and sockets
- ★ And custom UDFs

Qserv design

- ★ Relational database, spatially-sharded with overlaps
- ★ Map/reduce-like processing







Key Features

Scalable spherical geometry

- ★ 0/360 RA wrap around, pole distortion, convex polygons,
- ★ accurate distance computation, functions for distance (angle),
- ★ point-in-spherical-region tests (circle, ellipse, box, convex polygon)
- ★ Custom (HTM-based) UDFs https://github.com/wangd/scisql

Optimized spatial joins for neighbor queries, cross-match

- ★ Spherical partitioning with overlap
- ★ Director table, secondary index
- ★ Two-level, 2nd level materialized on-the-fly

Shared scans

- ★ Continuous, sequential scans through data, including L3 distributed tables
- ★ (Non-interactive) queries attached to appropriate running scan

All internal complexity transparent to end-users

Tests and demonstrations

Target for production

~500 nodes clusters in 2 international data-centers

Running now

Development platform (CC-IN2P3)

400 cores, 800 GB memory
500 TB storage,

★ **~35 TB data set** on 2*25 nodes
★ *Computing KPI on **~70TB dataset***

Prototype Data Access Center (NCSA)

500 cores, 4 TB memory
700 TB storage,
SDSS Stripe 82 and
Wide-Field Infrared Survey Explorer (WISE) data loaded

Scale testing to date @IN2P3

S15 large scale tests

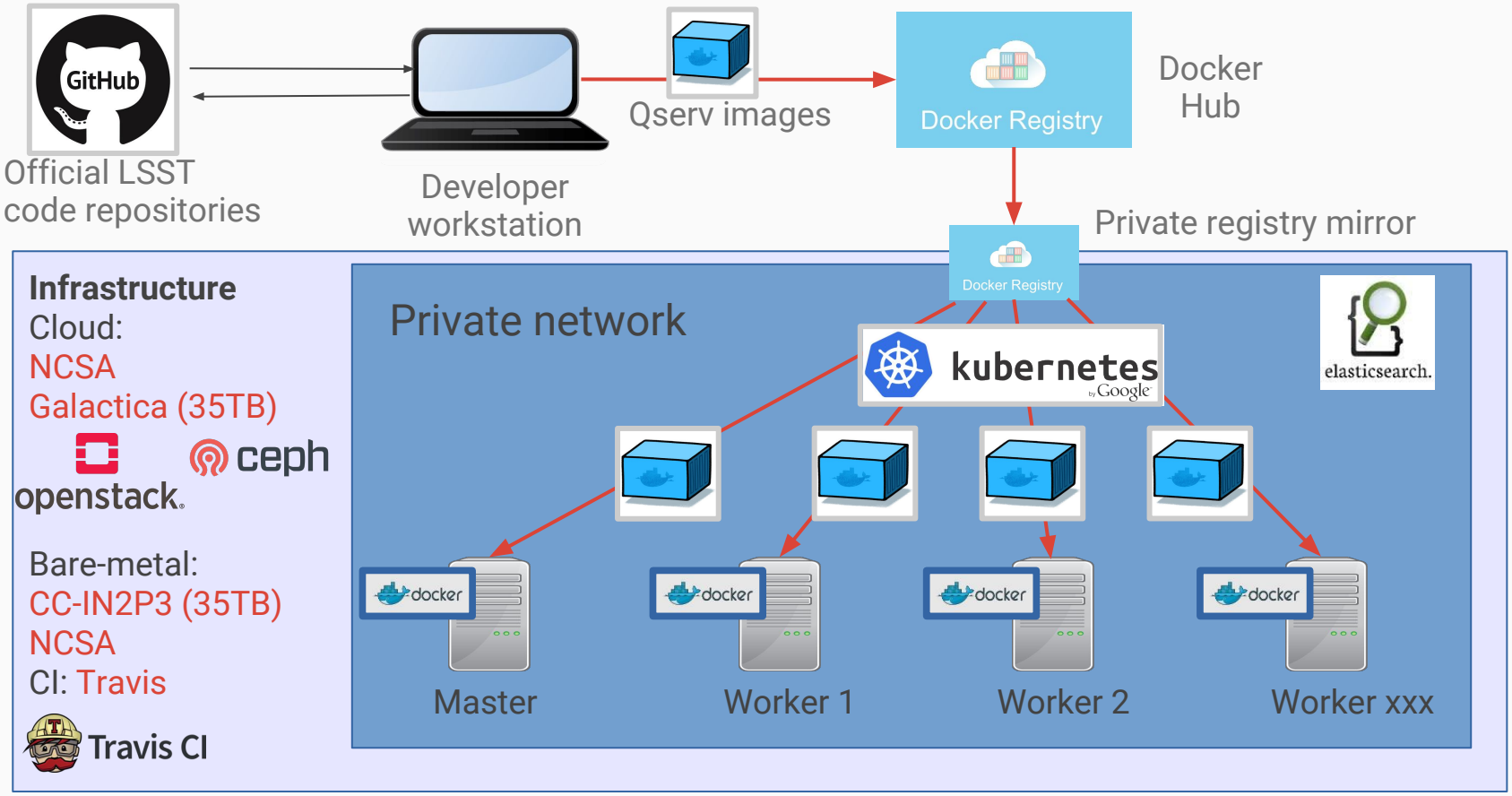
Data: replicated SDSS Stripe 82
~10% DR1 (~2B Object, ~35B Source, ~172B F. Source)

Hardware: 25 nodes @ IN2P3, 2 x 1.8GHz 4 core, 16G RAM

Simul. 50 low-volume queries + 5 high-volume queries:
<1s for low-volume queries
~15m for high-volume Object scans
~1h for Source scans
=> Promising performances

See S15 Large Scale Tests

Automated Qserv deployment



Summary

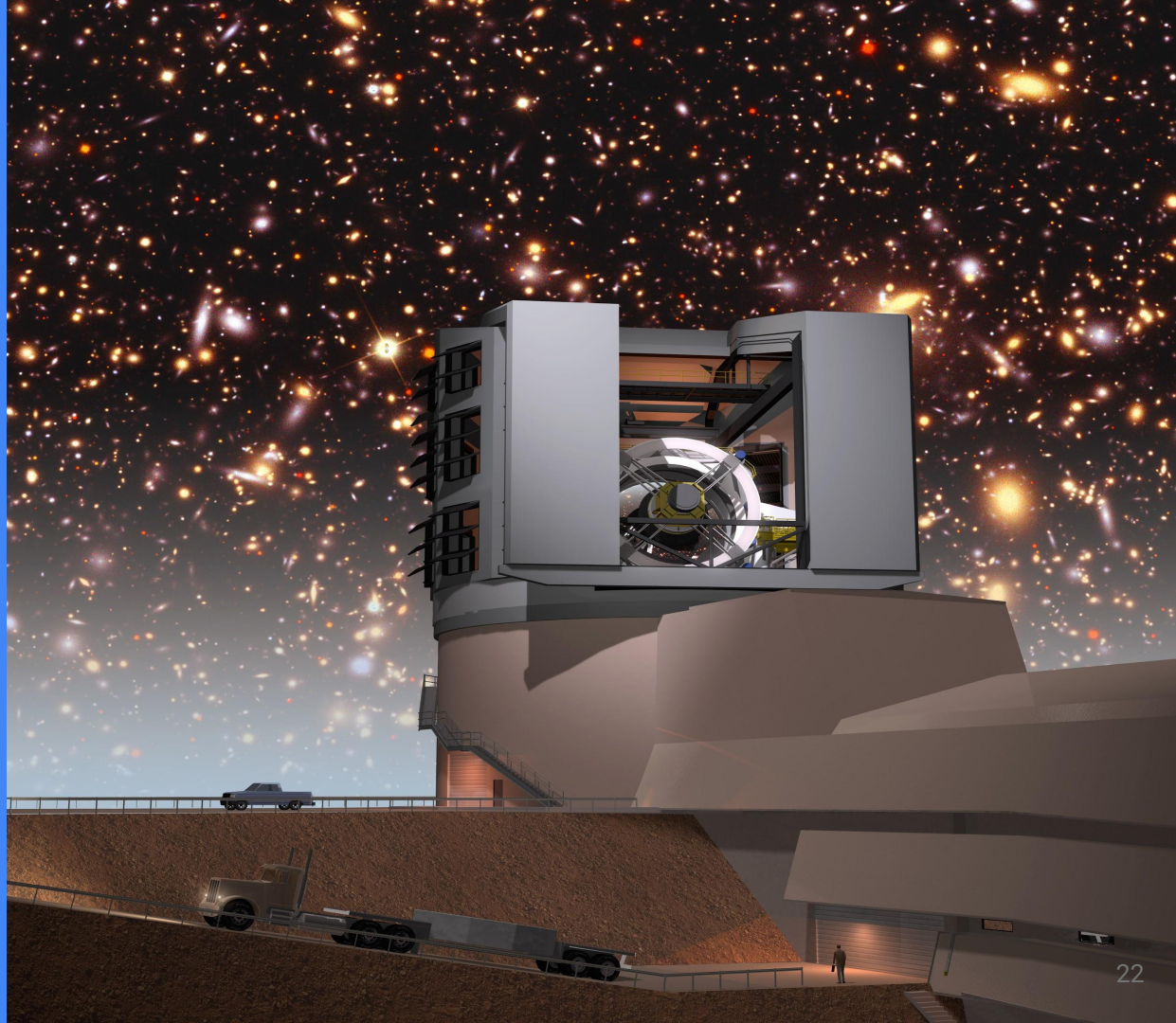
- ★ Big Data with Complex Analytics
- ★ Spatially-sharded, map/reduce-like RDBMS
- ★ Open source + custom glue
- ★ Optimized for astronomical data sets at scale
- ★ Have working prototype
- ★ Turning it into a production system
- ★ Want to learn more?
  - http://ls.st/4gh (Database Design doc)
  - http://ls.st/6ym (User Manual)
- ★ Are you an adventurous super early adopter? You can try it now
  - http://ls.st/89y (Qserv Documentation)

Thanks!

Contact:

Fabrice JAMMES
LPC
Clermont-Ferrand

fabrice.jammes@in2p3.fr



Implementation Details