A web portal to analyze and distribute cosmology data

https://cosmohub.pic.es

Outlook

- What is CosmoHub
- Motivation
- Drawbacks
- Hadoop solution
- Comparison
- Demo
- Conclusions & future work
Build your own Universe

Real-time data analysis of massive cosmological data without any SQL knowledge

Hundreds of millions of observed and simulated galaxies
Superfast queries mean superfast results
Features to make you work faster and easier
Online plotting preview and data download
CosmoHub was first thought as a way to share data within two closed related projects, the Physics of the Accelerating Universe (PAU) and the Marenostrum Institut de Ciències de l’Espai simulations (MICE)

- It was built on top of a PostgreSQL relational database
- It was developed by people from the Institut de Ciències de l’Espai (ICE), the Port d’Informació Científica (PIC - www.pic.es), CIEMAT and IFAE
- It was hosted and operated at PIC
CosmoHub is currently supporting four different cosmology projects:

- ~ 400 users
- ~ 1300 custom catalogs
- ~ 250 prebuilt downloads
- ~ 3 TiB hosted data
- > $10^9$ objects
Already available features

- Custom catalogs without any SQL knowledge (CSV.BZ2 only)
- Plot & preview tool: small sample of data using a scatter plot or generate a 1D-histogram (query time limited to < 2’)
- Value-Added-Data ready to be downloaded
What happened?

- MICECATv2.0 catalog contains about 500M entries with more than 120 fields
- Managing large volume of data in PostgreSQL had some drawbacks:
  - Indices are not used for large datasets
  - Most custom catalogs lasted several hours
- Changing the schema was very slow
- Removing large subsets of data is very inefficient
- Future galaxy catalogs will contain a few $10^9$ entries
Apache Hadoop & Hive

- Apache Hadoop:
  - one of the most popular Big Data platform
  - open-source software
  - based on commodity computer clusters
    - distributed storage and distributed processing
    - scalable from dozens up to even thousands of nodes
    - failure tolerance
- Apache Hive: query over massive data volumes
Hadoop basics

Input file

Node1
A
B
C
D
E

Node2
D
C
A
E
B

Node3
C
A
D
B
E

Node4
E
A
D
B
E

Node5
A

Hadoop vs. PostgreSQL

- Larger relative gain in execution time for increasing complexity in datasets and/or as queries request larger data volumes

(Comparisons are odious. It is very likely to unjust to one or other of them)

**Hadoop**
- Nodes: 15
- Cores: 12 (Intel Xeon X5650 @ 2.67 GHz) [180]
- RAM: 24 GiB [360 GiB]
- DISK: 1 TiB [15 TiB raw; ~5 TiB net]
- Network: 1 GbE

**PostgreSQL**
- Hardware:
  - Cores: 24 (Intel Xeon X5675 @ 3.07 GHz)
  - RAM: 96 GiB
  - DISK: 600 GB HDD x 8 (in RAID 6) ~ 3.6 TB net
  - Network 1GbE
- Software:
  - Scientific Linux 6.1
  - PostgreSQL 9.1
Hadoop vs. PostgreSQL

```
SELECT ra, dec, z, z_v, x_c, y_c, z_c FROM micecatv1 WHERE x_c < 700 AND y_c < 700 AND z_c < 700; (~5.8M out of ~205M rows)
```
Hadoop vs. PostgreSQL

SELECT x_c, y_c, z_c FROM micecatv1 WHERE x_c < 1e3 AND y_c < 1e3 AND z_c < 1e3; (~16.5M out of ~205M rows)
SELECT coadd_objects_id, ra, dec, mag_auto_i, magerr_auto_i desdm_zp, mean_z_bpz, z_mc_bpz FROM des_y1a1 WHERE modest_class = 1 AND flags_gold = 0 AND flags_badregion = 0;

(\textasciitilde81.9M \text{out of } \textasciitilde137M \text{rows})
Hadoop vs. PostgreSQL

```
SELECT ra_gal, dec_gal, kappa, gamma1, gamma2
FROM micecatv2 WHERE lmhalo >= 12.16 AND
flag_central = 0 AND z_cgal > 0.4 AND z_cgal < 0.6;
```

(~25.9M out of ~500M rows)

- **PostgreSQL**
  - Execution time (seconds): 5930
- **Hive**
  - Execution time (seconds): 102

‘Speedup’ 58.1
SELECT coadd_objects_id, ra, dec, mag_auto_g, mag_auto_r, mag_auto_i, mag_auto_z, mean_z_bpz, mode_z_bpz, median_z_bpz, z_mc_bpz, t_b, spread_model_i, spreaderr_model_i, modest_class FROM des_y1a1 WHERE mag_auto_i > 17.5 AND mag_auto_i < 22 AND (flags_badregion <= 3 and flags_gold = 0) AND ((mag_auto_g - mag_auto_r) BETWEEN -1. and 3.) AND ((mag_auto_r - mag_auto_i) BETWEEN -1. and 2.5) AND ((mag_auto_i - mag_auto_z) BETWEEN -1. and 2.) AND (ra < 15 or ra > 290 or dec < -35); (~34.8M out of ~137M rows)
Hadoop vs. PostgreSQL

```
SELECT coadd_objects_id, ra, dec, mag_auto_g, mag_auto_r, mag_auto_i, mag_auto_z, mean_z_bpz, mode_z_bpz, median_z_bpz, z_mc_bpz, t_b, spread_model_i, spreaderr_model_i, modest_class FROM des_y1a1 WHERE mag_auto_i > 17.5 AND mag_auto_i < 22 AND (flags_badregion <= 3 and flags_gold = 0) AND ((mag_auto_g - mag_auto_r) BETWEEN -1. and 3.) AND ((mag_auto_r - mag_auto_i) BETWEEN -1. and 2.5) AND ((mag_auto_i - mag_auto_z) BETWEEN -1. and 2.) AND (ra < 15 or ra > 290 or dec < -35);
```

PostgreSQL

Hive

<table>
<thead>
<tr>
<th>Execution time (seconds)</th>
</tr>
</thead>
<tbody>
<tr>
<td>15240</td>
</tr>
</tbody>
</table>

'Speedup' 100.0

(~34.8M out of ~137M rows)
SELECT z, log_m FROM micecatv1 WHERE z < 0.25 AND z > 0.23 AND ra < 20 AND dec < 20; (~52K out of ~205M rows)

Properly using indices and a very small amount of data requested!
- CosmoHub is a portal for real-time analysis and distribution of massive cosmology data without any SQL knowledge.

- It is built on top of Hadoop and uses the Apache Hive infrastructure.

- It is fully developed, hosted and operated at PIC.
New features

- Real time analysis (no time constraint)
- Sampling: select a random subset of the catalog to get faster results when exploring the data
- Heatmap plot
- 2 more file formats to download the selected data: FITS and ASDF
New features

- Real time analysis (no time constraint)
- Sampling: select a random subset of the catalog to get faster results when exploring the data
- Heatmap plot
- 2 more file formats to download the selected data: FITS and ASDF
New features

- Real time analysis (no time constraint)
- Sampling: select a random subset of the catalog to get faster results when exploring the data
- Heatmap plot
- 2 more file formats to download the selected data: FITS and ASDF
New features

- Real time analysis (no time constraint)
- Sampling: select a random subset of the catalog to get faster results when exploring the data
- Heatmap plot
- 2 more file formats to download the selected data: FITS and ASDF
Demo
Conclusions & future work

### Conclusions
- Great improvement in response time
- New release is more reliable
- Still exploring the vast Hadoop ecosystem

### Future work
- New plot types and analysis
- Collaboration with more experiments
  - More data, more catalogs, more users
  - Other use cases (other than Cosmology)
Thanks for your attention!

https://cosmohub.pic.es
Hive tuning

- We have set the platform so that queries over large tables are really fast:
  - Apache Tez execution engine instead of the venerable Map-reduce engine
  - ORCfile: a new table (column based) storage format
  - Vectorized query technique: batches of 1024 rows at once
Load balancing

- Set up two different queues given the two different profiles:
  - ‘Interactive’: real-time analysis (low latency)
  - ‘Batch’: custom catalogs (high latency)

- Configure queue shares and preemption:
  - batch jobs take idle resources to maximize efficiency (10-90)
  - interactive jobs can take resources from batch queue (90-100)
Backend

- ReST API powered by Flask:
  - flask-restful - ReST framework
  - sqlalchemy - database ORM
  - websockets - bidirectional communications
  - gevent - asynchronous framework
  - pyhive - hive connection library
  - pyhdfs - hdfs bindings
Responsive Web interface powered by:

- Angular JS - web app oriented HTML framework
- Bootstrap - responsive frontend framework
- Plot.ly for plotting
- Wordpress as backend to edit "static" content
Demo

CosmoHub YouTube channel