COSMO HUB on Hadoop

https://cosmohub.pic.es

A web portal to analyze and distribute cosmological data

J.Carretero, P.Tallada, J.Casals, M.Caubet, C.Neissner, N.Tonello, J.Delgado, F.Torradeflot, M.Delfino, S.Serrano, P.Fosalba

PIC
port d'informació
científica

ICE CSIC - IEEC

IFAE
25 years 1991-2016
Institut de Física d'Altes Energies

EXCELENCIA SEVERO OCHOA

BIST Barcelona Institute of Science and Technology

MINISTERIO DE ECONOMÍA Y COMPETITIVIDAD

Ciemat
Centro de Investigaciones Energéticas, Medioambientales y Tecnológicas

EXCELENCIA MARÍA DE MAEZTU

Outline

- What is CosmoHub
- How it started
- Hadoop platform
- Evaluation
- Some facts
- Demo
- Conclusions & future work

PIC
port d'informació
científica

COSMO HUB

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 means superfast results

Features to make you work faster and easier

Online plotting preview and data download

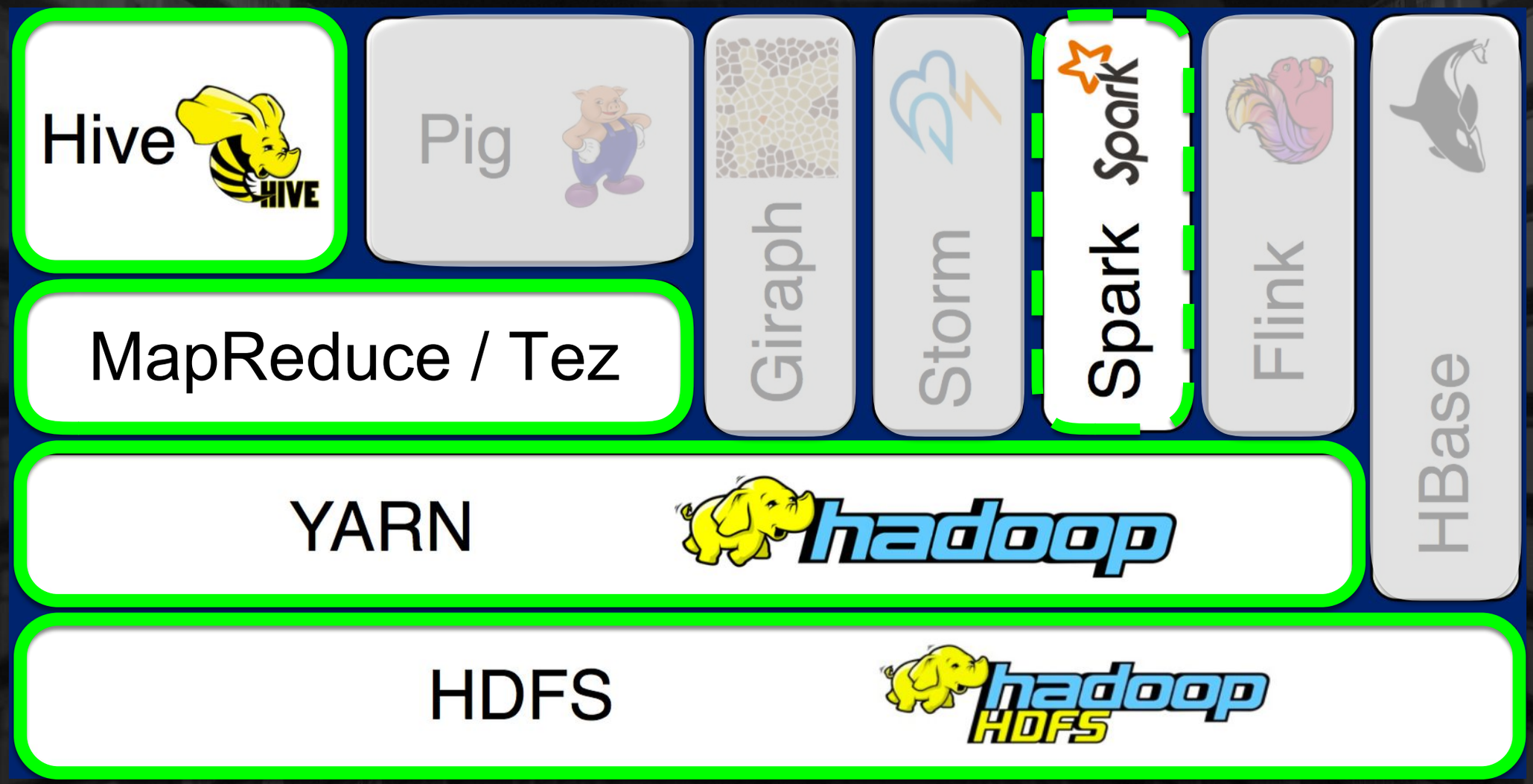
How it started?

- CosmoHub was created to share data from two cosmology projects
- It was built on top of a PostgresQL relational DB
- PostgreSQL wasn’t scaling how we wanted as data volume grew
  - Indices were not used in large datasets
  - Most queries lasted several hours
  - Changing the schema and removing data became very slow and inefficient
- We knew we would grow even more…

Apache Hadoop + Hive

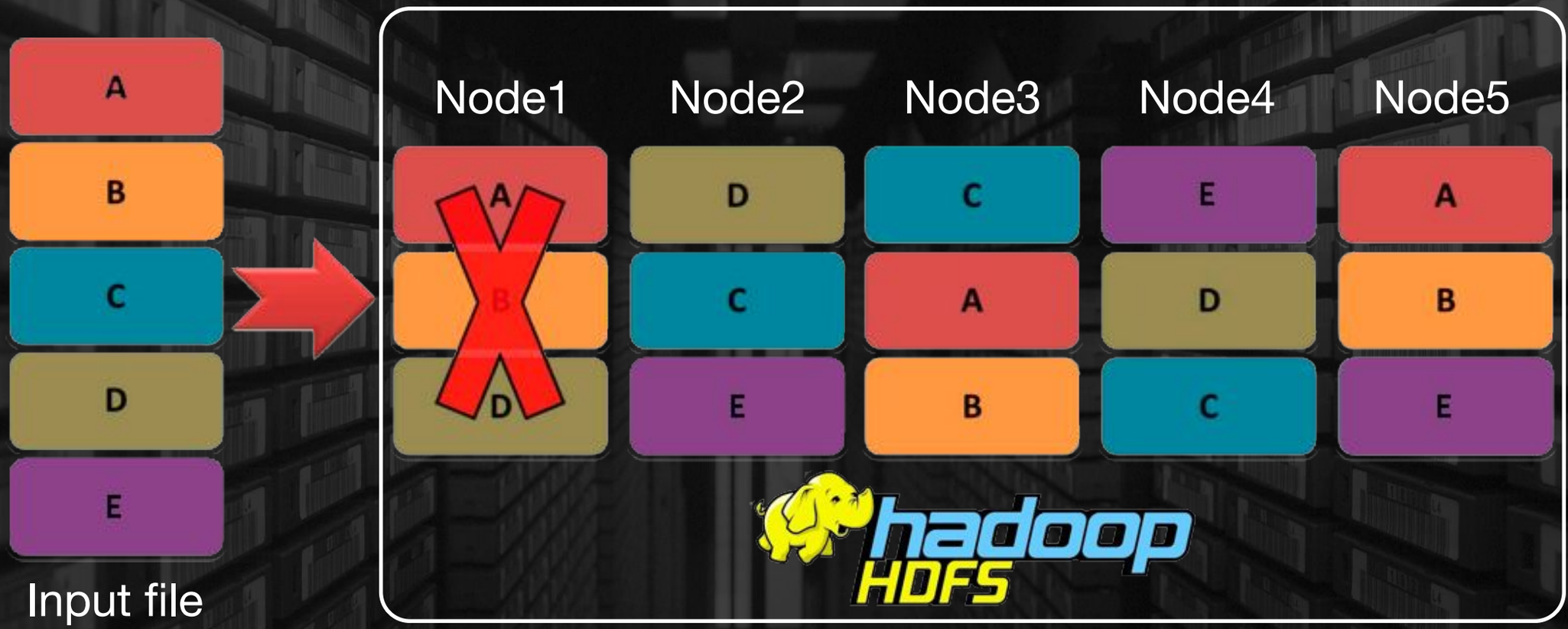
- Apache Hadoop
  - One of the most popular Big Data platforms
  - Open source
  - Distributed storage and processing
    - Based on commodity computer clusters
      - ✓ May use old/refurbished machines → **cheaper**
    - Scalable from dozens up to thousands of nodes
    - Failure tolerance
- Apache Hive
  - Query over massive data volumes using SQL

Hadoop stack



HORTONWORKS®

Hadoop HDFS


A
B
C
D
E
Input file
Node1
Node2
Node3
Node4
Node5
A
B
D
C
D
C
A
E
D
B
E
B
C
A
E
B
C
D
E
hadoop
HDFS

PostgreSQL vs Hadoop

Postgresql (9.1)  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

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: 1GbE

PostgreSQL vs Hadoop

```
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)

```
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)

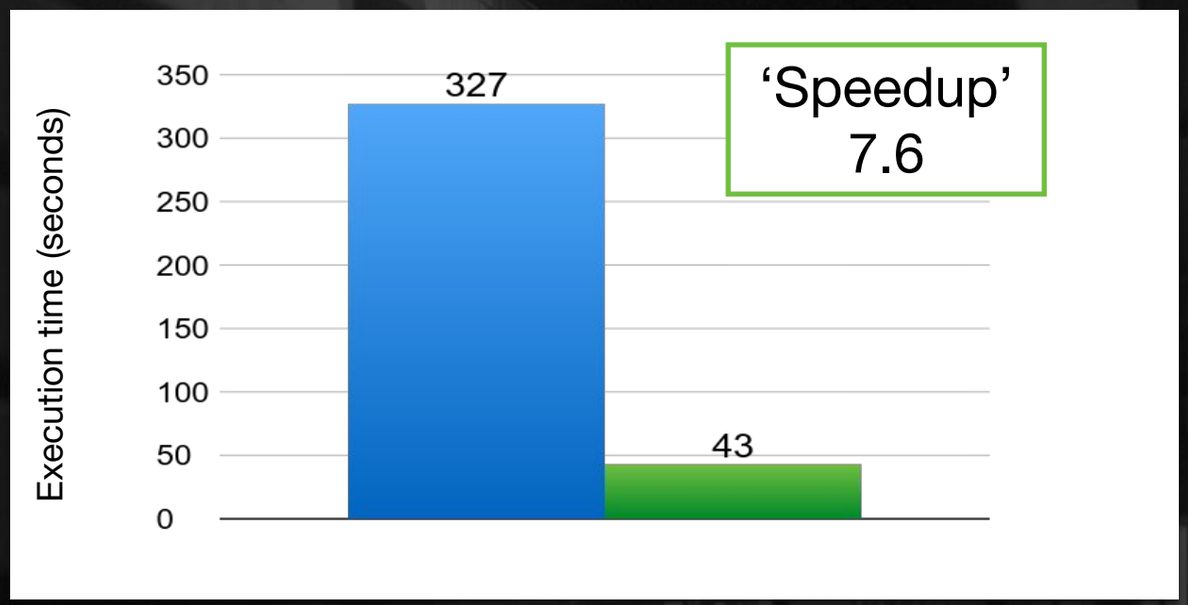
```
SELECT ra_gal, dec_gal, kappa,
gamma1, gamma2
FROM micecatv2
WHERE lmhalo >= 12.16 AND
flag_central = 0 AND z_cgal > 0.4
AND z_cgal < 0.6;
```

(~25.9M out of ~500M rows)

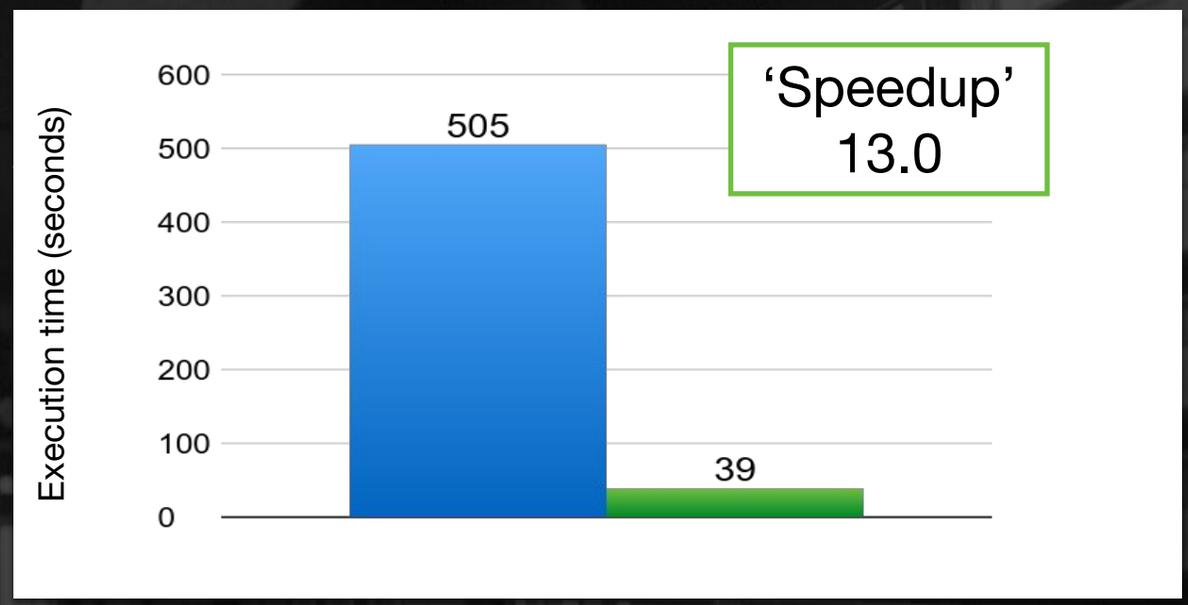
```
SELECT coadd_objects_id, ra, dec, mag_auto_g,
mag_auto_r, mag_auto_i, mag_auto_z, mean_z_bpz,
mode_z_bpz, median_z_bpz, z_mc_bpz, t_b,
spread_model_i, spreaderr_model_i, modest_class
FROM des_y1a1 WHERE mag_auto_i > 17.5 AND
mag_auto_i < 22 AND (flags_badregion <= 3 and
flags_gold = 0) AND ((mag_auto_g - 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)

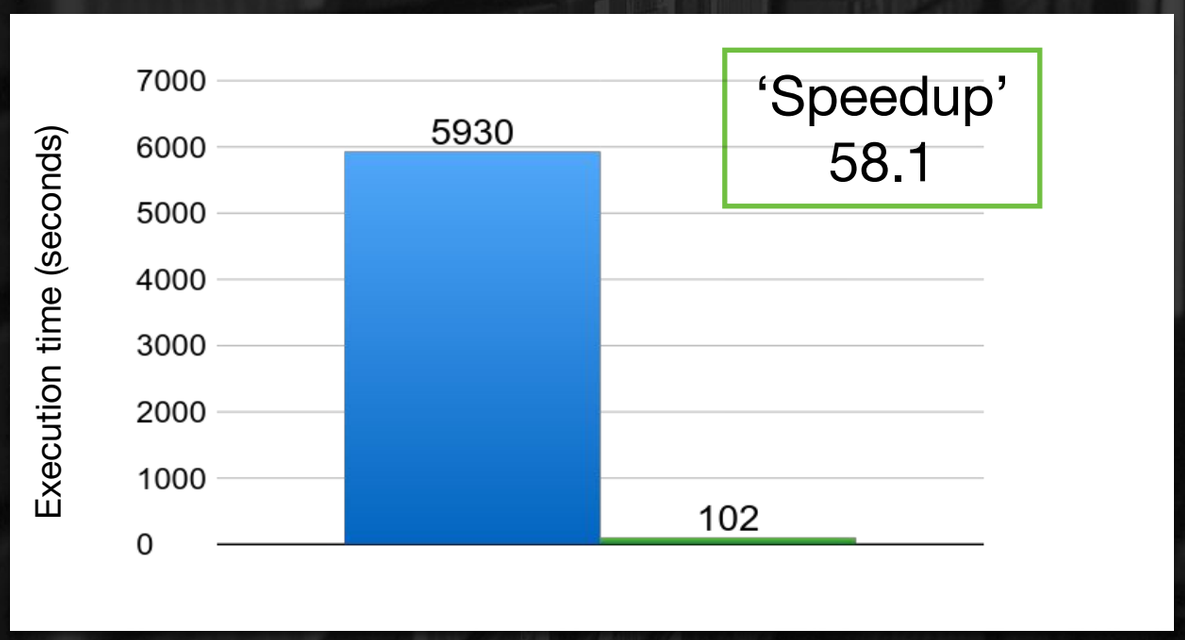
PostgreSQL vs Hadoop



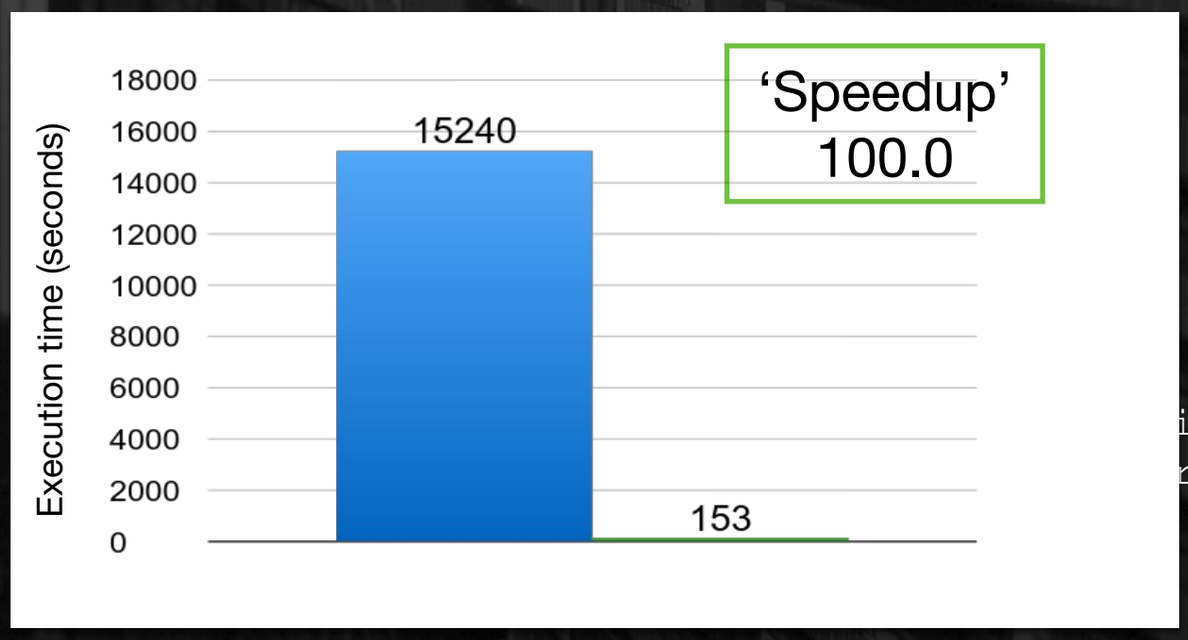
(~5.8M out of ~205M rows)



(~16.5M out of ~205M rows)



(~25.9M out of ~500M rows)



(~34.8M out of ~137M rows)

PostgreSQL vs Hadoop

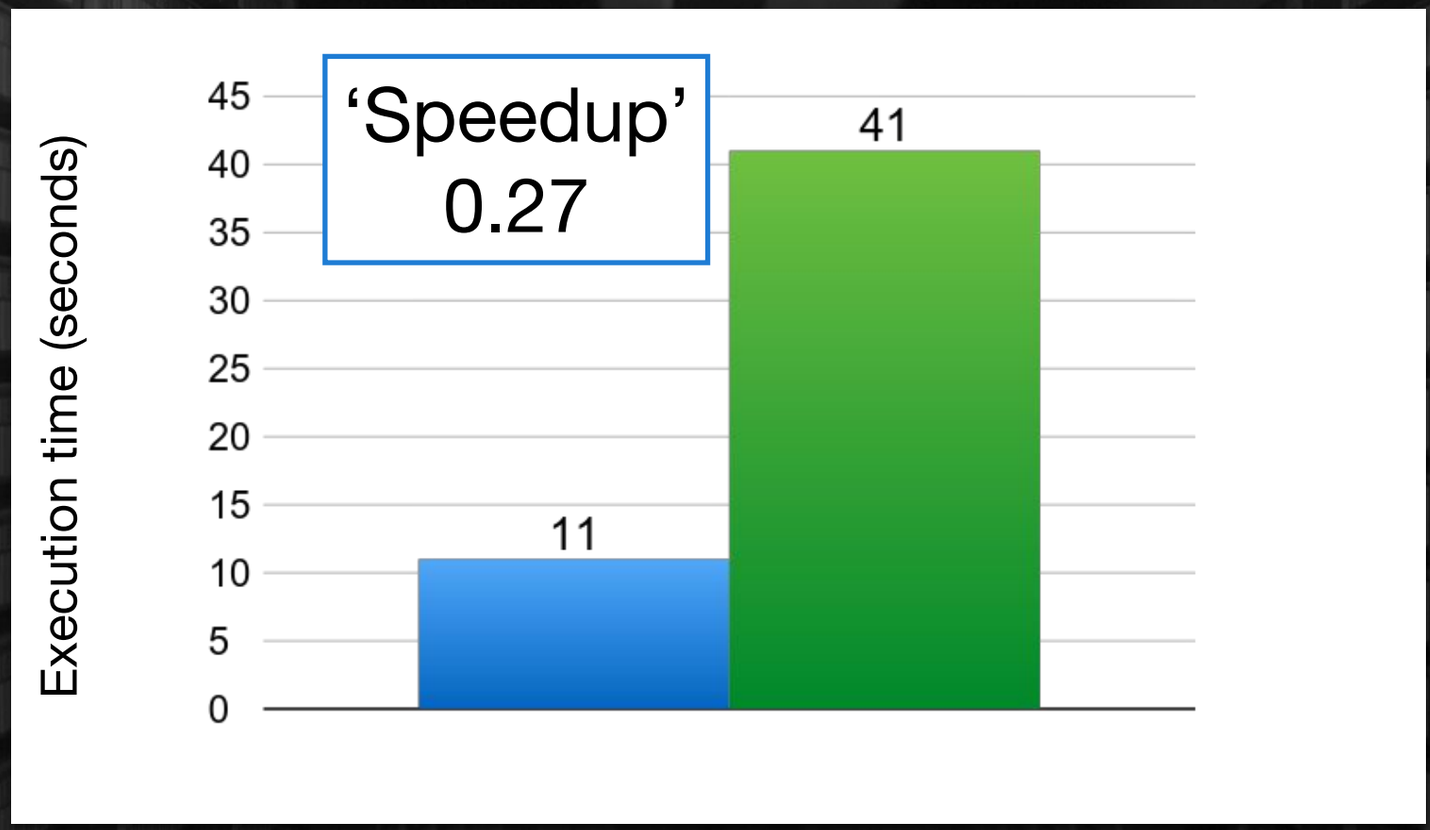
```
SELECT z, log_m FROM micecatv1 WHERE z < .25
AND z > .23 AND ra < 20 AND dec < 20;
```

(~52K out of ~205M rows)

Properly using indices and a very small amount of data requested!

PostgreSQL vs Hadoop

```
SELECT z, log_m FROM micecatv1 WHERE z < .25
AND z > .23 AND ra < 20 AND dec < 20;
```



(~52K out of ~205M rows)

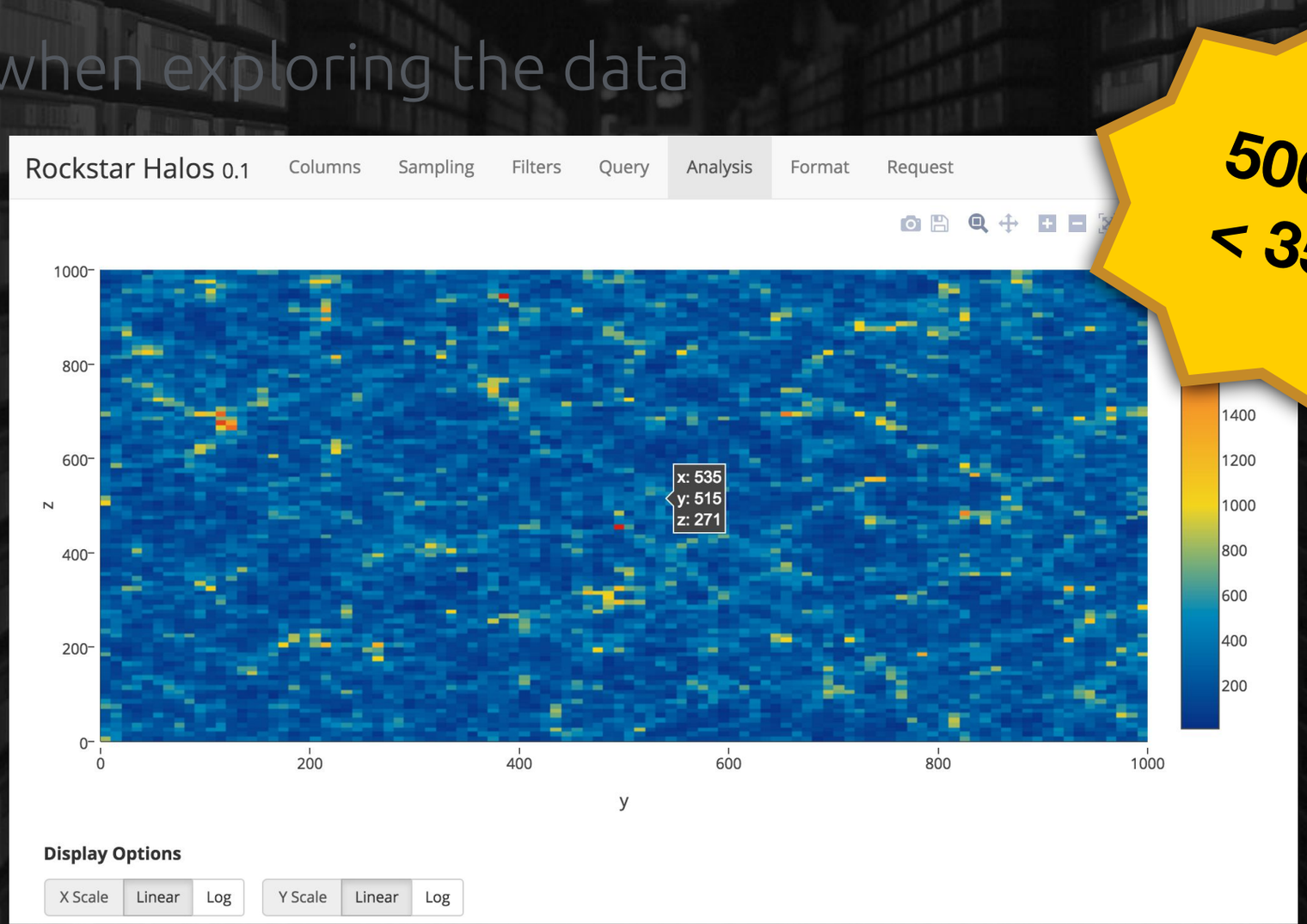
Properly using indices and a very small amount of data requested!

CosmoHub on Hadoop

- Easy scalability with old servers
- Reliability after possible failures
- Impressive performance
  - A faster system than the previous version
  - CosmoHub switched focus from batch catalog generation to **interactive catalog analysis**

CosmoHub on Hadoop

- **Interactive analysis**
  - Histogram and heatmap plots
- Sampling: select a random subset of the catalog to get faster results when exploring the data



500M
< 35”

CosmoHub on Hadoop

- Interactive analysis (no time constraint)
  - Histogram and heatmap plots
- Sampling: select a random subset of the catalog to get faster results when exploring the data

Some facts

~ 450 users **~ 1500** custom catalogs **~ 6 TiB** hosted data **> 10^{10}** objects

Some of the experiments in CosmoHub

Demo

Click me!

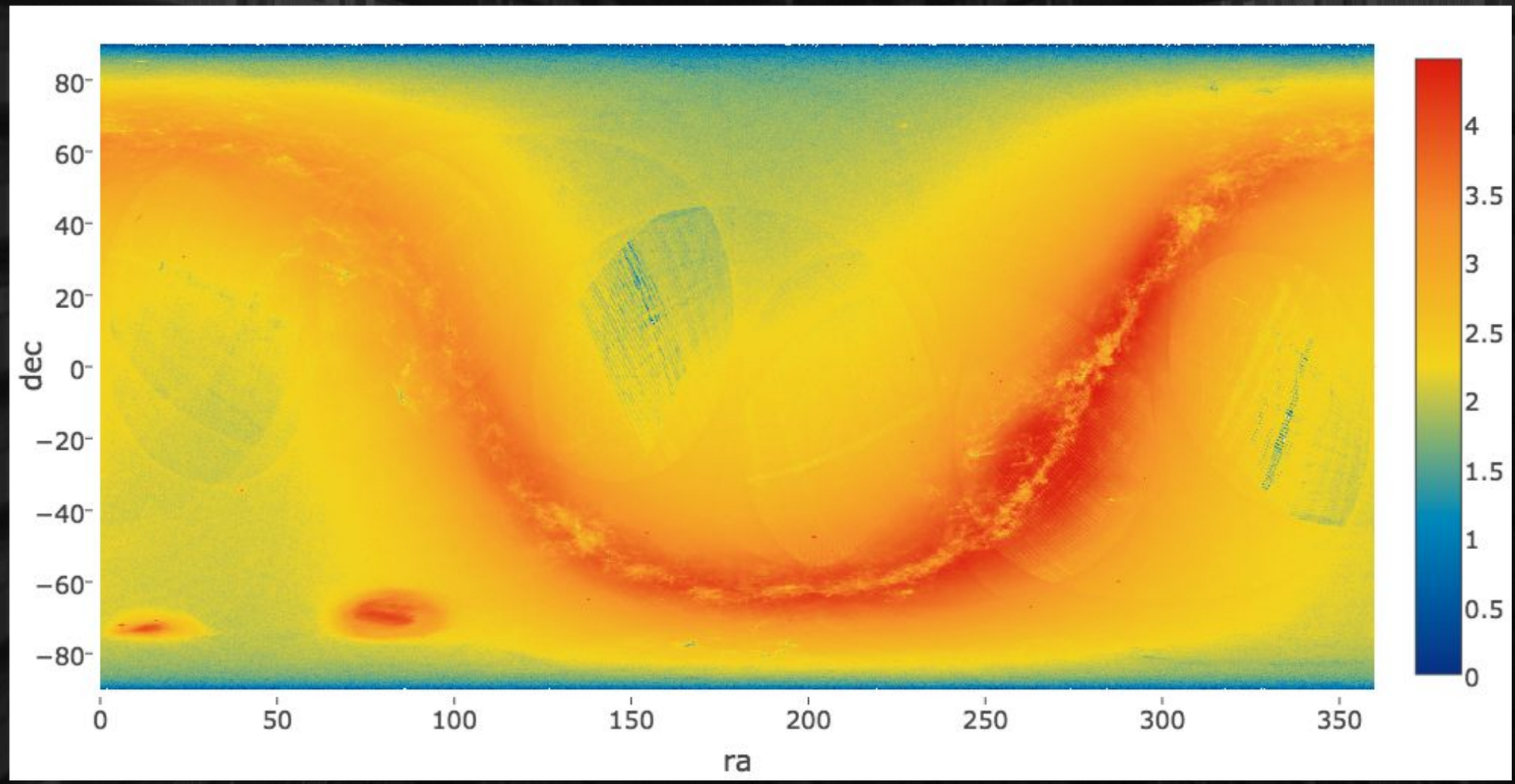
Video demo...

Demo

Full Gaia catalog: ~1.1×10^9

ra, dec distribution

1000x1000 bins in ~ 4’



Conclusions & future work

- New release
  - Great improvement in response time
  - More reliable
  - New plot types and analysis
- More data, more catalogs, more users
- Other use cases (other than Cosmology)
- Still exploring the vast Hadoop ecosystem

COSMO HUB on Hadoop

… is public

… is free

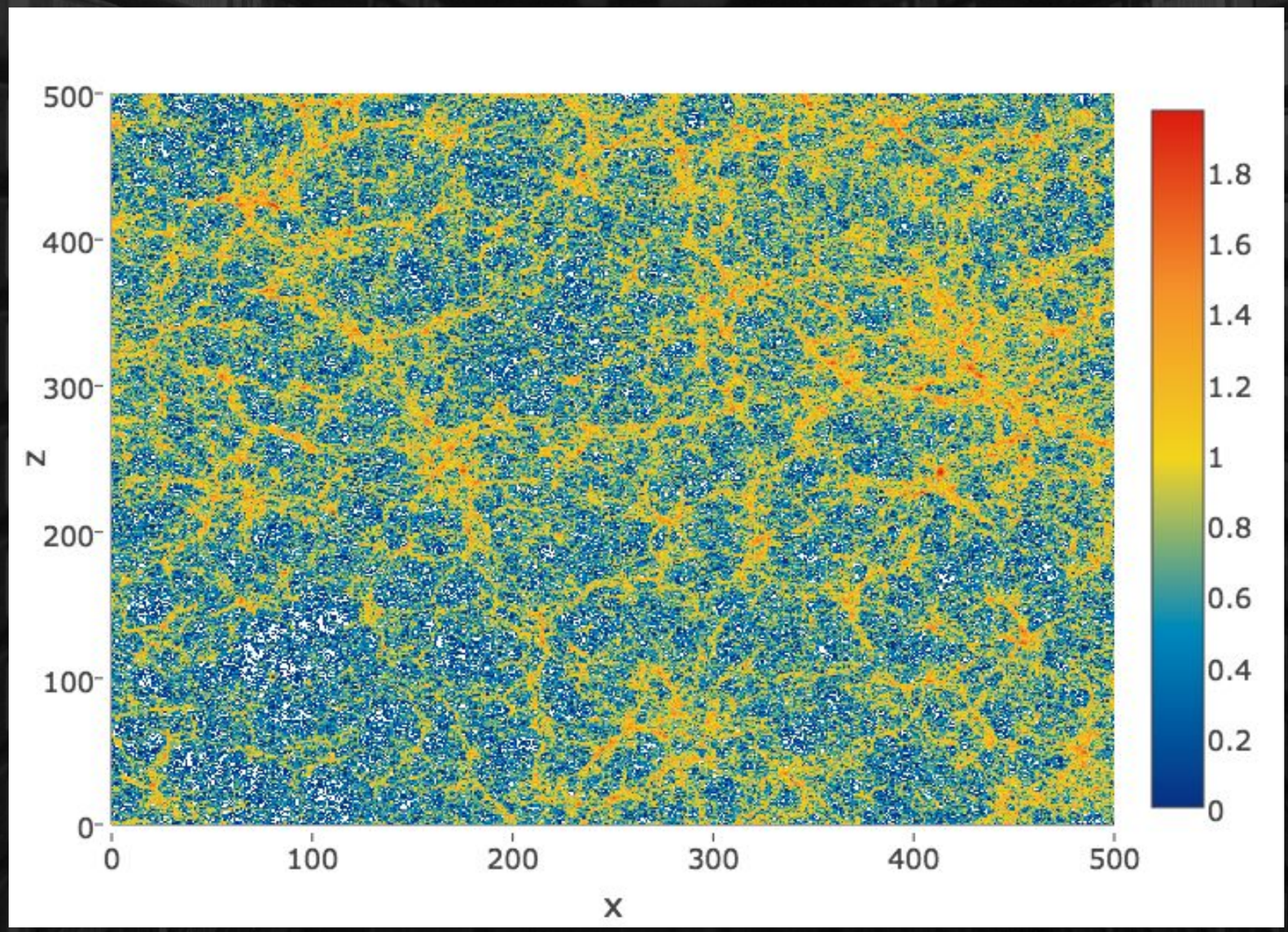
… is open to data from more projects

https://cosmohub.pic.es
@Cosmo_Hub
cosmohub@pic.es

Backup slides

Demo

Euclid Flagship dark matter halo catalog: ~5.5x109
500x50x500 Mpc/h section
1000x1000 bins in ~ 4’



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

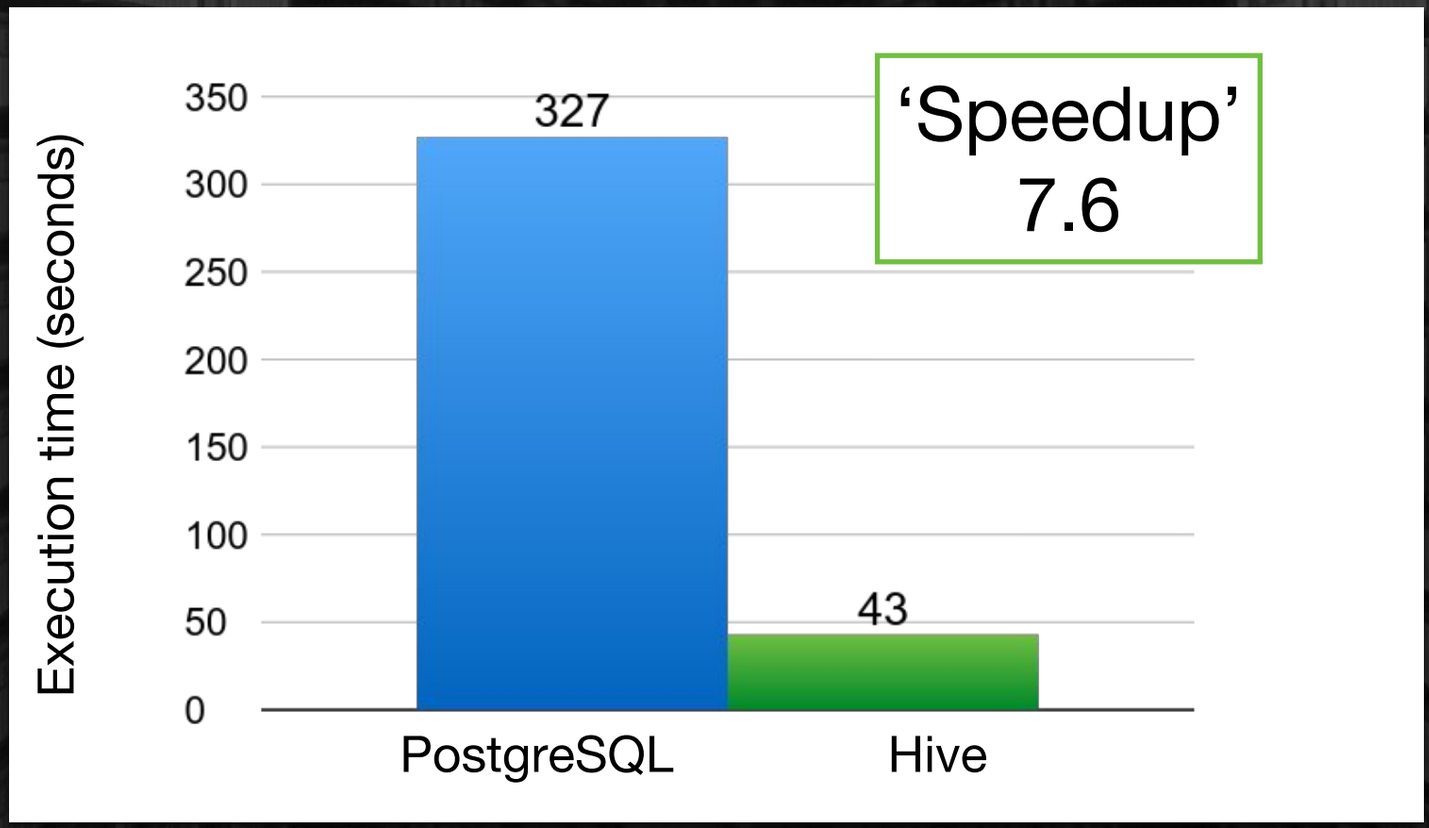
- ReST API powered by Flask:
  - flask-restful - ReST framework
  - sqlalchemy - database ORM
  - websockets - bidirectional communications
  - gevent - asynchronous framework
  - pyhive - hive connection library
  - pyhdfs - hdfs bindings

Frontend

- 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

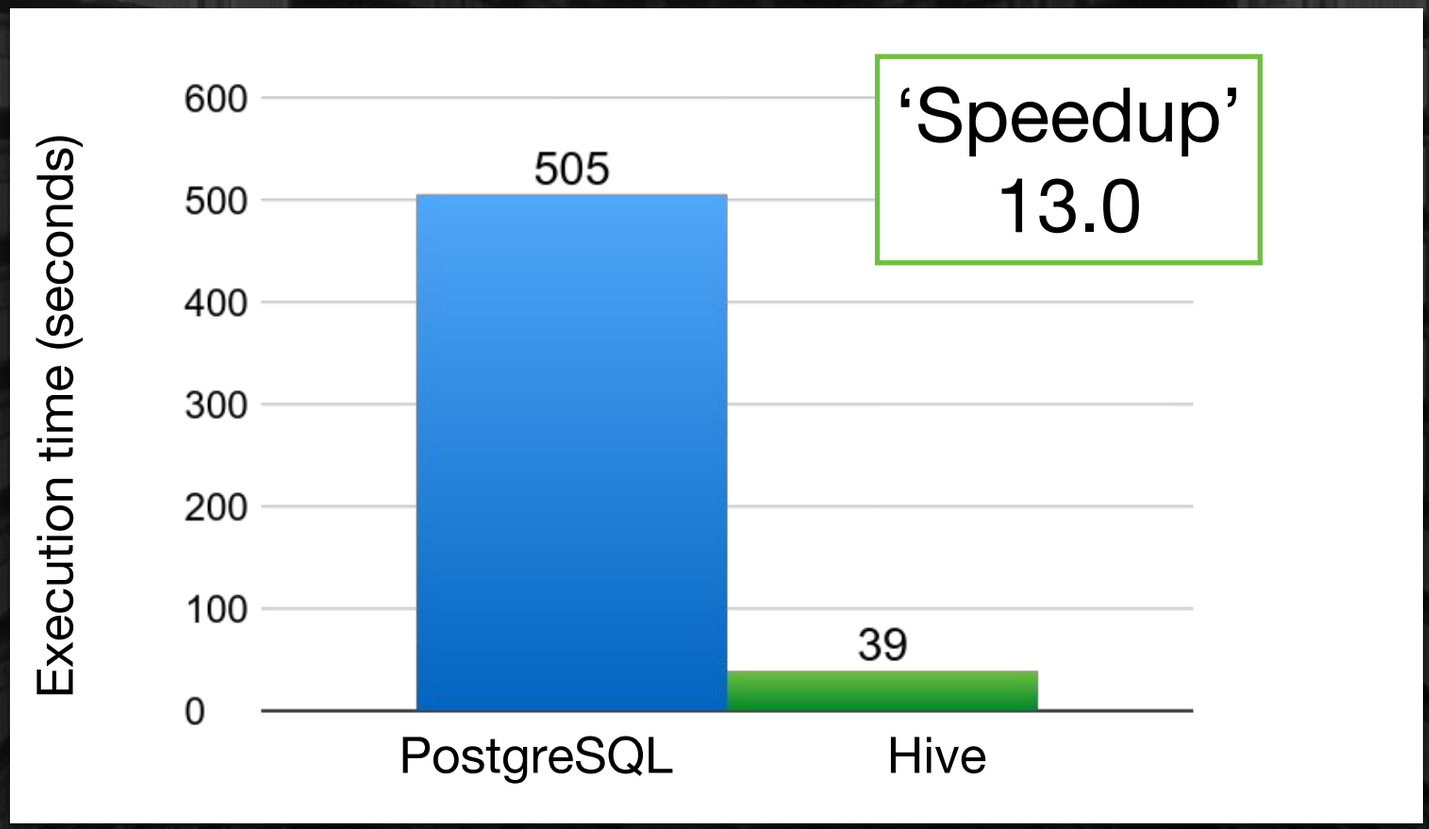
PostgreSQL vs Hadoop

```
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)



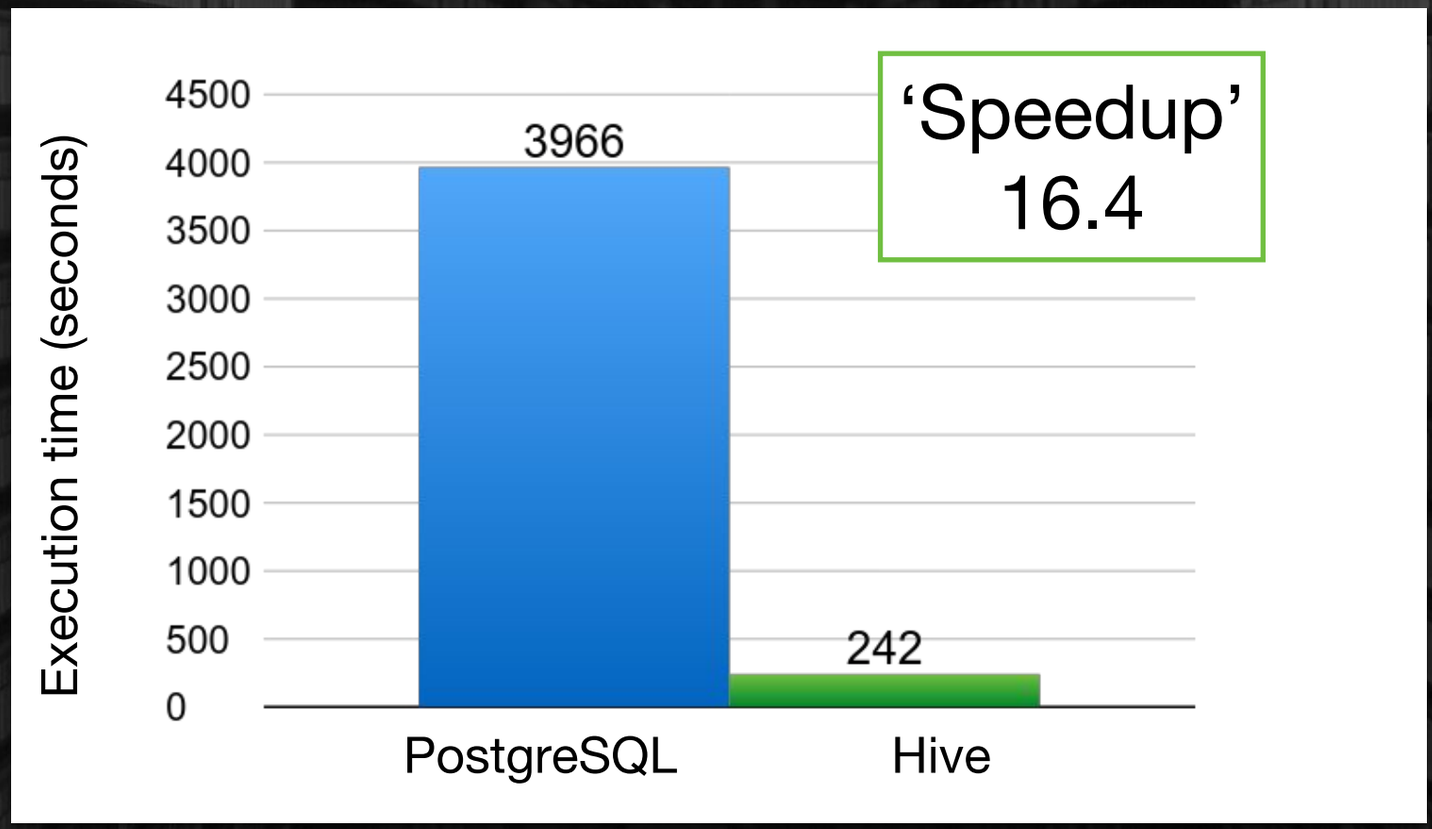
PostgreSQL vs Hadoop

```
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)



PostgreSQL vs Hadoop

```
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;
```
(~81.9M out of ~137M rows)



PostgreSQL vs Hadoop

```
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)