UNIVERSITÉ DE GENÈVE
FACULTÉ DES SCIENCES
Département d'astronomie

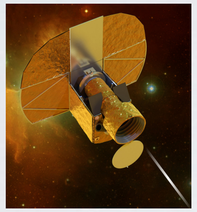
OBSERVATOIRE DE GENÈVE
FONDÉ EN 1772

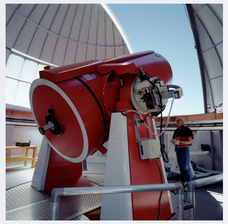
PlanetS
National Centre of Competence in Research

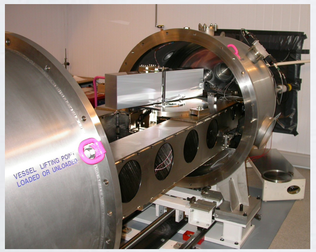
NoSQL Databases

An efficient way to store and query heterogeneous astronomical data in DACE

Nicolas Buchschacher - University of Geneva - ADASS 2018

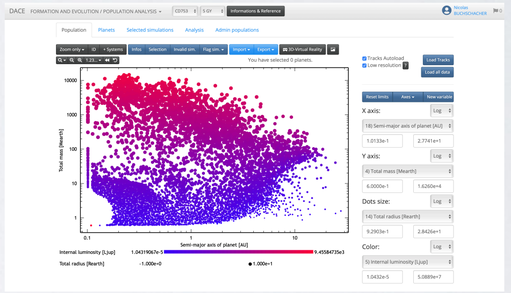


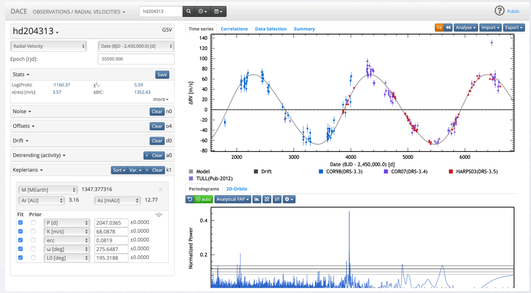




Solr

cassandra





UNIVERSITÉ DE GENÈVE
FACULTÉ DES SCIENCES
Département d'astronomie

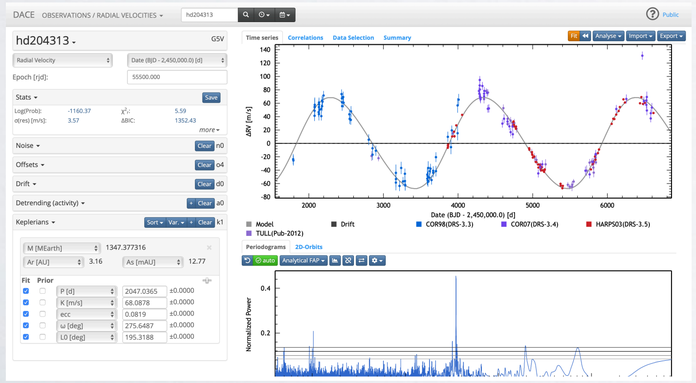
OBSERVATOIRE DE GENÈVE FONDÉ EN 1772

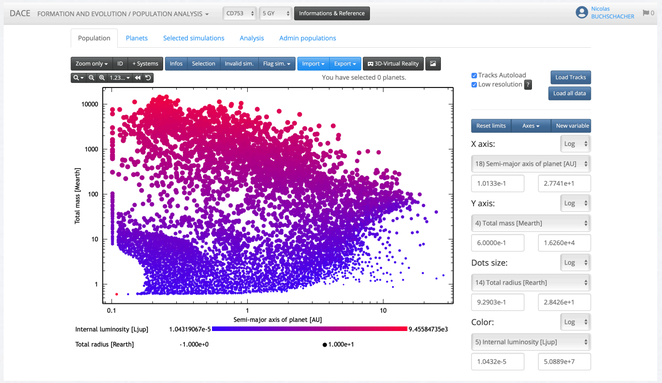
PlanetS
National Centre of Competence in Research

DACE

https://dace.unige.ch

- **D**ata and **A**nalysis **C**enter for **E**xoplanets.
- Facility to store, exchange and analyse data related to exoplanets (observations and synthetic populations).
- Web front end and python API to query the database (in dev).
- **3D visualisation** (come to the demo booth !)





The Requirements

- Store heterogeneous observational data produced by different instruments.
- Regularly adapt the data model (we are at the end of the chain).
- Store synthetic population simulations data points (~500M / Pop).
- Ensure high availability (load balancing) and persistence (replications).

CASSANDRA

- Fully distributed database developed by Apache.
- Column oriented storage => can have thousand columns in a table.
- High availability (no Master-Slave). Every node is equivalent.
- Asynchronous and automatic replication across the nodes based on a replication factor.
- Consistency level can be controlled for each query.

CAP (Brewer) Theorem

This is impossible for a distributed database to simultaneously provide more than 2 of the 3 following requirements:

- **Consistency**: Every read receives the most recent value, or an error.
- **Availability**: Every read or write request receives a non-error reply.
- **Partition tolerance**: The system continues to work properly if there are some missing nodes.

=> **Cassandra** is considered as an **AP** database. But consistency can be managed with the "consistency level" in each query.

CASSANDRA (Storage)

Row Oriented (SQL)

ID	Param1	Param2	Param3
1	v 11	v 21	NULL
2	v 12	NULL	NULL
3	v 13	NULL	v 33
4	v 14	v 24	NULL

Column Oriented

ID	Param1
1	v 11
2	v 12
3	v 13
4	v 14

ID	Param2
1	v 21
4	v 24

ID	Param3
3	v 33

CASSANDRA (Storage)

Row Oriented (SQL)

ID	Param1	Param2	Param3	Param4
1	v 11	v 21	NULL	NULL
2	v 12	NULL	NULL	NULL
3	v 13	NULL	v 33	NULL
4	v 14	v 24	NULL	NULL
5	v 15	NULL	v 35	v 45

Column Oriented

ID	Param1
1	v 11
2	v 12
3	v 13
4	v 14
5	v 51

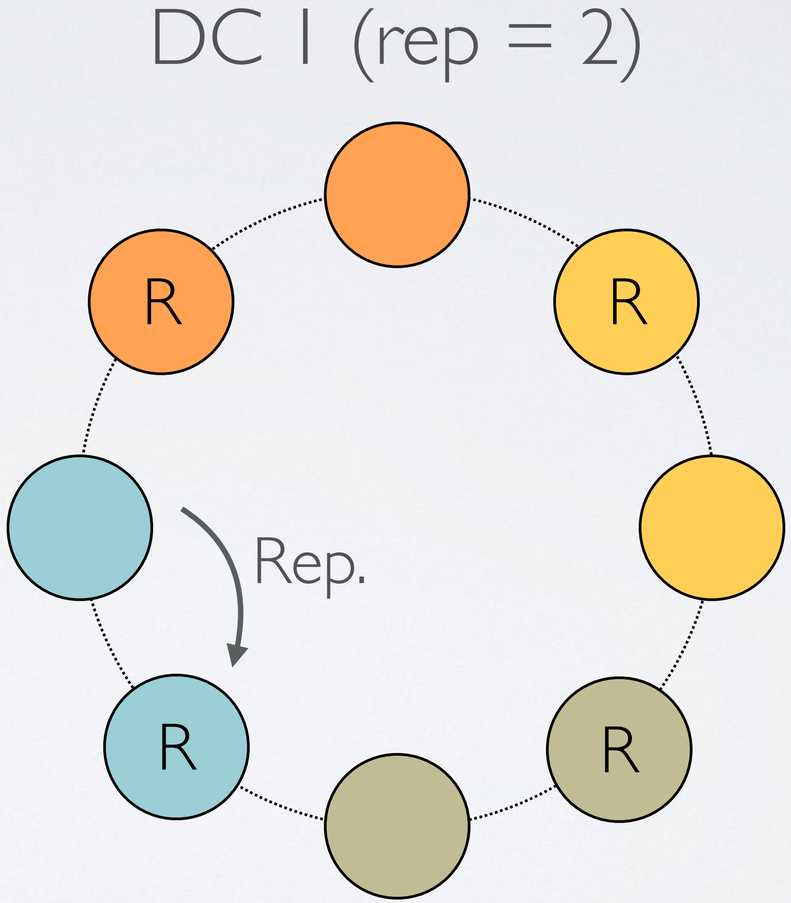
ID	Param2
1	v 21
4	v 24

ID	Param3
3	v 33
5	v 35

ID	Param4
5	v 45

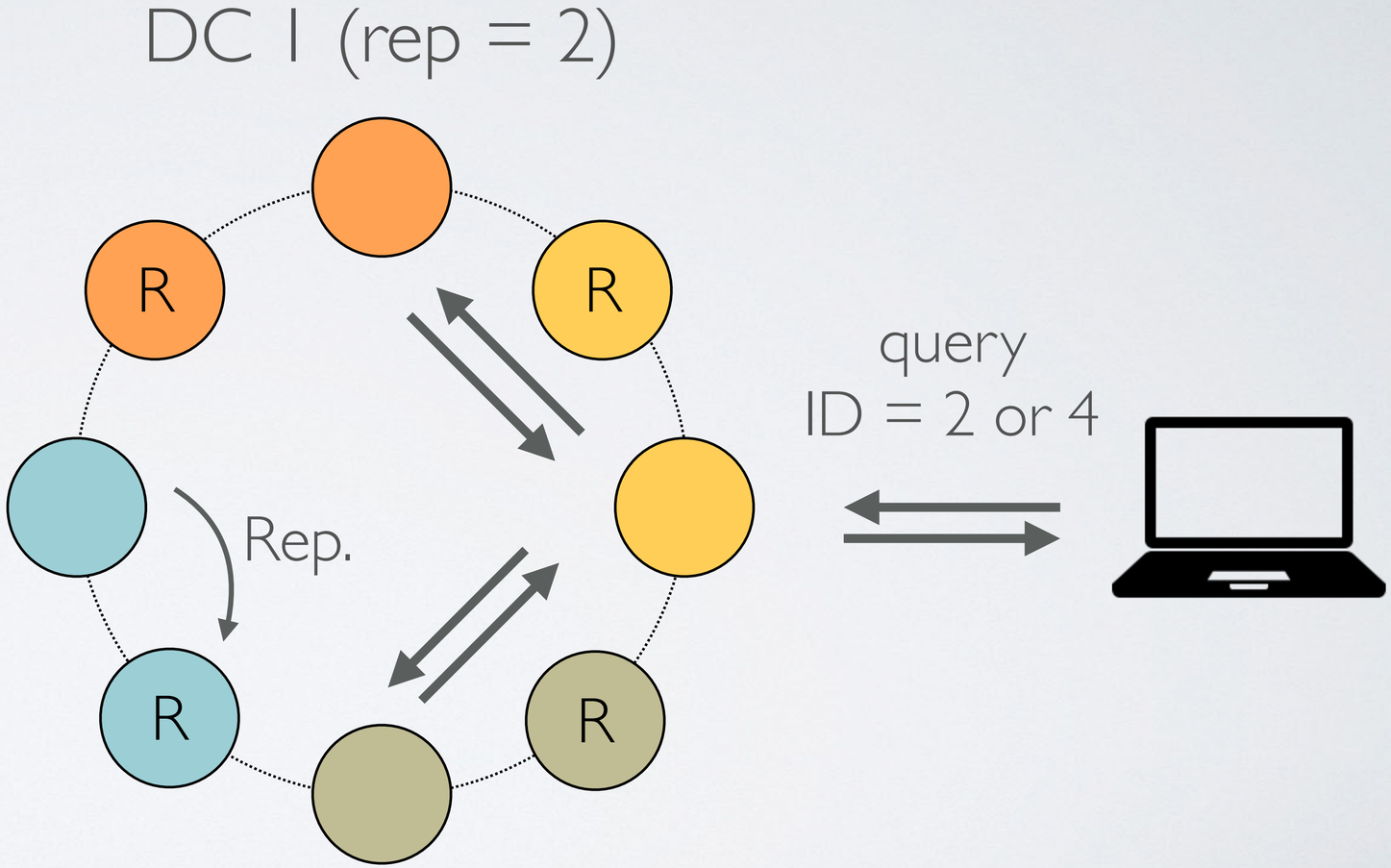
CASSANDRA Partitions & Replication

ID	Param1	Param2	Param3	Param4
1	v 11	v 21	NULL	v 41
2	v 12	NULL	NULL	NULL
3	v 13	NULL	v 33	v 43
4	v 14	v 24	NULL	v 44

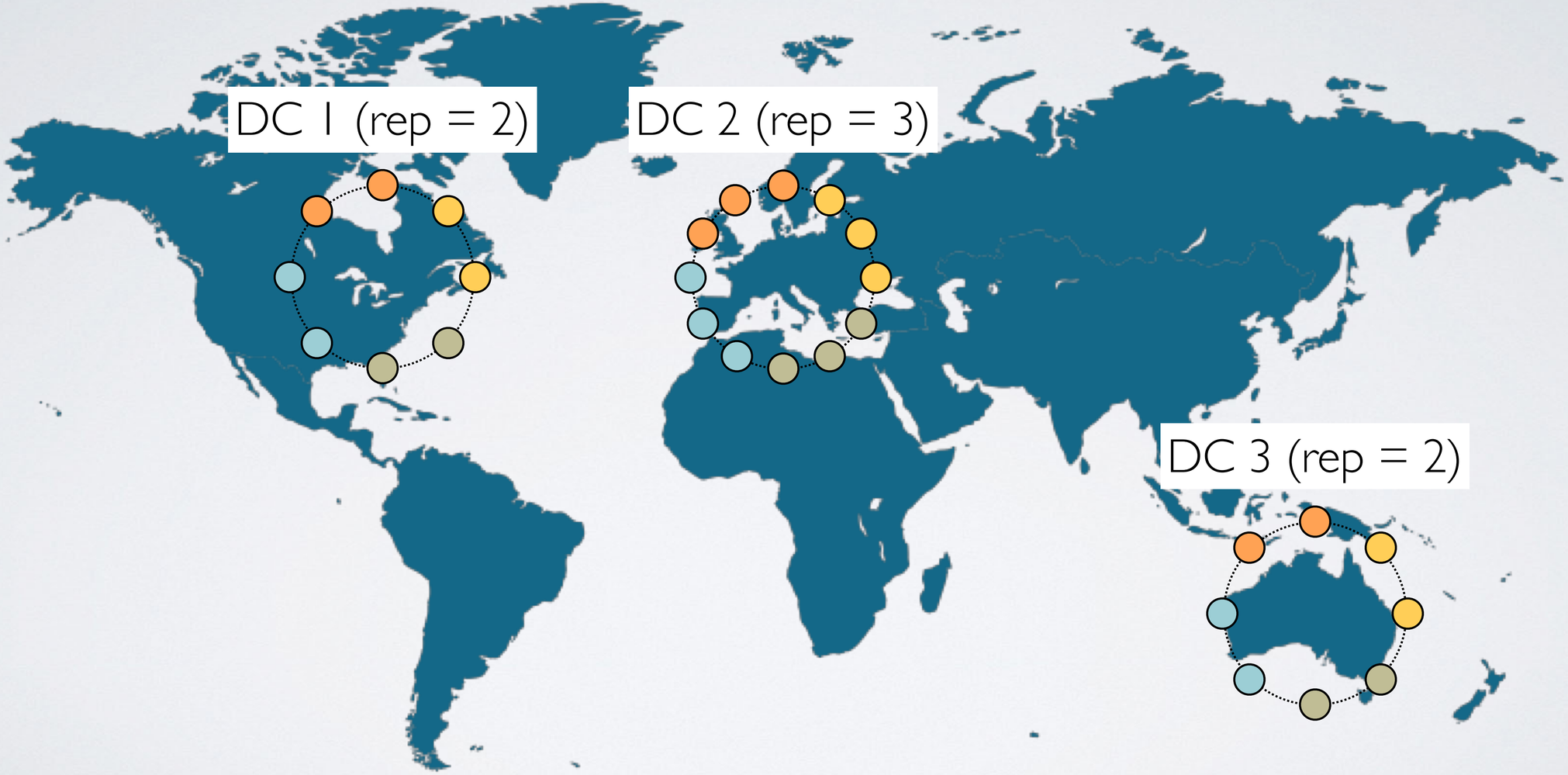


CASSANDRA Partitions & Replication

ID	Param1	Param2	Param3	Param4
1	v 11	v 21	NULL	v 41
2	v 12	NULL	NULL	NULL
3	v 13	NULL	v 33	v 43
4	v 14	v 24	NULL	v 44



CASSANDRA Partitions & Replication



CASSANDRA Pros & Cons

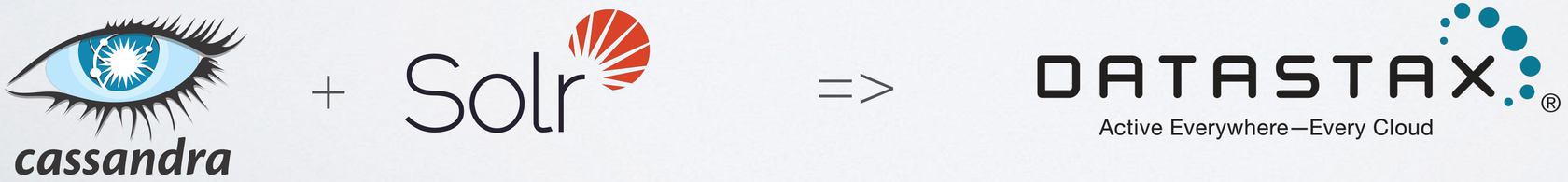
+ Store a lot of heterogenous columns without performance impact.
+ Fully distributed (no Master-Slave): all nodes are equivalent.
+ Easily scalable by adding new nodes.
+ Open source, compatible with major operating systems.
+ Easily expandable to the PB scale (Apple ex: 75'000 nodes).

- No relation between tables => No JOIN operations.
- Not transactional (not important in our case).
- Poor set of search and filter operations.

Solr / SolrCloud

- Open source enterprise search platform.
- Powerful indexer (full-text search, spatial, filtering, sort …) based on Apache Lucene.
- REST API with a lot of supported formats (JSON, CSV, Python,…).
- SolrCloud is the distributed version of Solr.

DSE combine both in a single software … but it's not free !!!



SQL vs NoSQL comparisons

SQL	NoSQL
Fixed schema, avoid column / table manipulations	Flexible schema, easily add new columns / parameters
Vertical scalability. Increase CPU, RAM and disks	Horizontal scalability (add nodes)
JOIN operations	No JOIN, no subqueries
Transactional operations	Not transactional
Centralised approach. Load balance using Master-Slave	De-centralised approach. Naturally load balanced
Long history and experience (1970)	New generation (~2000)

Conclusion

- Is NoSQL better ? **No !!!** It's different …
- Consider NoSQL databases if you need to store heterogeneous data and want a flexible data model.
- NoSQL is Big Data oriented.
- Consider NoSQL databases if availability and scalability is a strong requirement.
- SQL is still on the stage and very powerful (we still use Postgres)
- Are VO standards ready for NoSQL ?

Thank you