RDBMS Systems

14 November 2017

Ignacio Coterillo
Computer Engineer, CERN

Market status

DB Engines Ranking (https://db-engines.com/en/ranking)

337 systems in ranking, November 2017

Rank			DBMS	Database Model	Score		
Nov 2017	Oct 2017	Nov 2016			Nov 2017	Oct 2017	Nov 2016
1.	1.	1.	Oracle	Relational DBMS	1360.05	+11.25	-52.96
2.	2.	2.	MySQL	Relational DBMS	1322.03	+23.20	-51.53
3.	3.	3.	Microsoft SQL Server	Relational DBMS	1215.08	+4.76	+1.27
4.	4.	4.	PostgreSQL	Relational DBMS	379.92	+6.64	+54.10
5.	5.	5.	MongoDB	Document store	330.47	+1.07	+5.00
6.	6.	6.	DB2	Relational DBMS	194.06	-0.53	+12.61
7.	7.	↑ 8.	Microsoft Access	Relational DBMS	133.31	+3.86	+7.34
8.	8.	↓ 7.	Cassandra	Wide column store	124.21	-0.58	-9.76
9.	9.	9.	Redis	Key-value store	121.18	-0.87	+5.64
10.	10.	↑ 11.	Elasticsearch	Search engine	119.41	-0.82	+16.84

RDBMS

Not typically thought of when planning new big data systems

Very probable to end having a big data problem when the original use case grows in time (either in size or scope)

Scaling Up: Buying Hardware

- Typically bigger/faster CPU's and more memory

```
The CPU count is typically used as reference in commercial license pricing
```

Oracle Engineered Systems (https://www.oracle.com/engineered-systems/exadata/database-machine-x7/index.html)

- Optional memory expansion to 384 GB (12 x 32 GB)
- 10 GBase-T or 25 GbE SFP+ public network
- 2 small form-factor front NVMe SSDs for data storage
- Comprehensive fault detection and notification
- Up to 912 CPU cores and 28.5 TB memory per rack for database processing
- Up to 360 CPU cores per rack dedicated to SQL processing in storage
- From 2 to 19 database servers per rack

One 10 Core Intel® Xeon® Processors

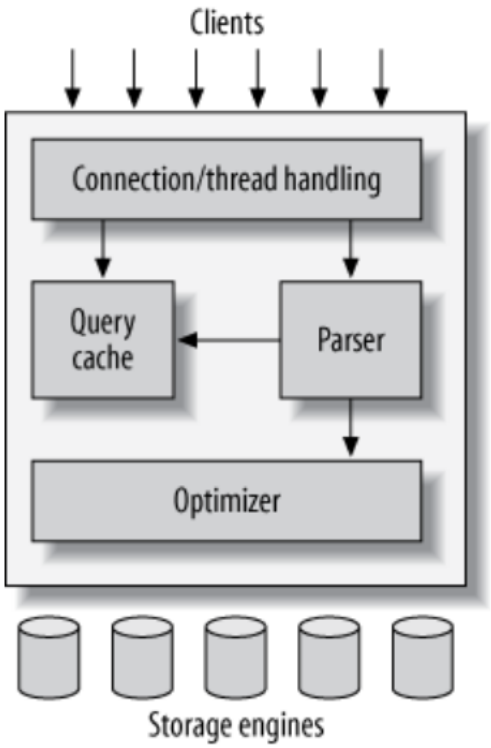
Up to 384 GB (12 x 32 GB) of main memory

Up to 12.8 TB of high-bandwidth NVME Flash for data storage

Tech: Storage Engines (1/2)

Most RDBMS are architecturally engineered in a modular way to support different storage systems (storage plugins).

MySQL Architecture (https://www.safaribooksonline.com/library/view/high-performance-mysql/9781449332471/ch01.html)



Tech: Storage Engines (2/2)

MySQL

- InnoDB: Default engine, transactional, ACID, MVCC
- Archive: For "Offline" data
- RocksDB: KV Store, embeddable, available as MyRocksDB (Facebook)
- TokuDB: For write heavy applications (reducing write amplification). Uses Fractal Indexes

Tech: Indexing

Indexes are data structures enabling quick lookup and access to information

Good indexing improve performance greatly

Too many indexes are counter-productive for Write Operations

```
Each index is an individual data structure which needs to be managed by the RDBMS. Each
new insert, delete, on a table will cause equivalent operations in the table indexes.
```

Tech: Covering Indexes (1/2)

```
mysql dod_mysql@dbod-fts3:newfts3lb> describe t_job;
+----------------------+-----------------------------------------------------------------------------------------------------------+---------+-------+-----------+---------+
| Field                | Type                                                                                                      | Null    | Key   |   Default | Extra   |
|----------------------+-----------------------------------------------------------------------------------------------------------+---------+-------+-----------+---------|
| job_id               | char(36)                                                                                                  | NO      | PRI   |    <null> |         |
| job_state            | enum('STAGING','SUBMITTED','READY','ACTIVE','FINISHED','FAILED','FINISHEDDIRTY','CANCELED','DELETE')      | NO      |       |    <null> |         |
| job_type             | char(1)                                                                                                   | YES     |       |    <null> |         |
| cancel_job           | char(1)                                                                                                   | YES     |       |    <null> |         |
| source_se            | varchar(255)                                                                                              | YES     | MUL   |    <null> |         |
| dest_se              | varchar(255)                                                                                              | YES     |       |    <null> |         |
| user_dn              | varchar(1024)                                                                                             | YES     |       |    <null> |         |
| cred_id              | char(16)                                                                                                  | YES     |       |    <null> |         |
| vo_name              | varchar(50)                                                                                               | YES     | MUL   |    <null> |         |
| reason               | varchar(2048)                                                                                             | YES     |       |    <null> |         |
| submit_time          | timestamp                                                                                                 | YES     | MUL   |    <null> |         |
| priority             | int(11)                                                                                                   | YES     |       |         3 |         |
| submit_host          | varchar(255)                                                                                              | YES     |       |    <null> |         |
| max_time_in_queue    | int(11)                                                                                                   | YES     |       |    <null> |         |
| space_token          | varchar(255)                                                                                              | YES     |       |    <null> |         |
| internal_job_params  | varchar(255)                                                                                              | YES     |       |    <null> |         |
| overwrite_flag       | char(1)                                                                                                   | YES     |       |    <null> |         |
| job_finished         | timestamp                                                                                                 | YES     | MUL   |    <null> |         |
| source_space_token   | varchar(255)                                                                                              | YES     |       |    <null> |         |
| copy_pin_lifetime    | int(11)                                                                                                   | YES     |       |    <null> |         |
| checksum_method      | char(1)                                                                                                   | YES     |       |    <null> |         |
| bring_online         | int(11)                                                                                                   | YES     |       |    <null> |         |
| retry                | int(11)                                                                                                   | YES     |       |         0 |         |
| retry_delay          | int(11)                                                                                                   | YES     |       |         0 |         |
| job_metadata         | text                                                                                                      | YES     |       |    <null> |         |
+----------------------+-----------------------------------------------------------------------------------------------------------+---------+-------+-----------+---------+
```

Tech: Covering Indexes (2/2)

```
PRIMARY KEY (`job_id`),
KEY `idx_vo_name` (`vo_name`),
KEY `idx_jobfinished` (`job_finished`),
KEY `idx_link` (`source_se`,`dest_se`),
KEY `idx_submission` (`submit_time`,`submit_host`)
```

Tech: Fractal Indexes

www.percona.com/doc/percona-tokudb/ft-index.html (https://www.percona.com/doc/percona-tokudb/ft-index.html)

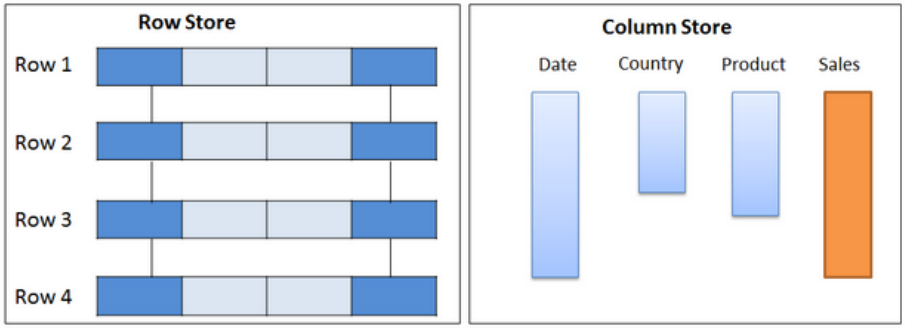
Fractal Indexes nodes contain buffers to store insertions temporally and wait until the buffer is complete before flushing to disk in a more efficient way.

Tech: Columnar oriented storage

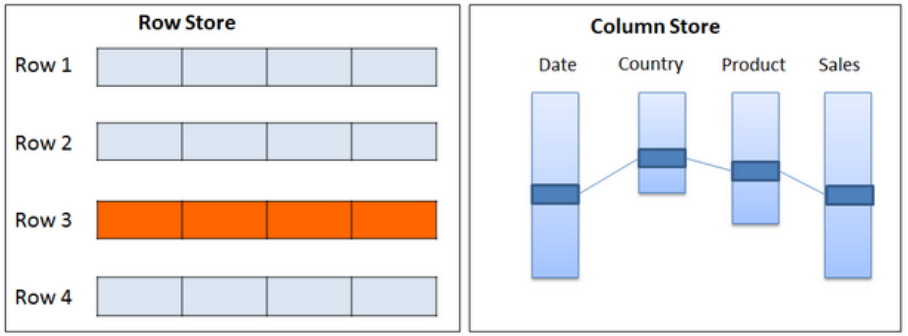
Table - SALES

	Date	Country	Product	Sales
Row 1	2013-01-01	India	Chocolate	1000
Row 2	2013-01-10	India	Ice-cream	2000
Row 3	2013-02-20	Germany	Chocolate	4000
Row 4	2013-03-01	US	Noodle	500

Column Operation: SELECT SUM(SALES) FROM SALES WHERE DATE > 2012-01-01



Row Operation: SELECT * FROM SALES WHERE COUNTRY = 'INDIA'



Tech: Partitions (1/2)

PostgreSQL Reference (https://www.postgresql.org/docs/10/static/ddl-partitioning.html)

```
CREATE TABLE measurement (
  city_id         int not null,
  logdate         date not null,
  peaktemp        int,
  unitsales       int
) PARTITION BY RANGE (logdate);

CREATE TABLE measurement_y2007m11 PARTITION OF measurement
  FOR VALUES FROM ('2007-11-01') TO ('2007-12-01')

CREATE TABLE measurement_y2007m12 PARTITION OF measurement
  FOR VALUES FROM ('2007-12-01') TO ('2008-01-01')
  TABLESPACE fasttablespace;

CREATE TABLE measurement_y2008m01 PARTITION OF measurement
  FOR VALUES FROM ('2008-01-01') TO ('2008-02-01')
  TABLESPACE fasttablespace
  WITH (parallel_workers = 4);
```

Tech: Partitions (2/2)

```
sys@ACCLOG:SQL> select table_name, partition_count from all_part_tables where partition_count
 > 200 order by partition_count;

TABLE_NAME                     PARTITION_COUNT
------------------------------ ---------------
DATA_VECTORSTRING                        1432
DATA_MATRIXNUMERIC                        1432
DATA_BEAM_MODES                         2624
DATA_FUNDAMENTAL                        3521
DATA_VECTORNUMERIC                        3522
DATA_STRING                            3522
DATA_NUMERIC_STATUS                        3524
DATA_NUMERIC                            3525
DATA_FUNDAMENTAL                        3904
DATA_NUMERIC_STATUS                        3939
DATA_VECTORNUMERIC                        3939
DATA_STRING                            3939
DATA_NUMERIC                            3939
```

Tech: Tablespaces (1/3)

Filesystem locations registered as logical units in the database which can be used to store data.

The basic way for extending the total storage available for a database system.

In most cases requires manual management

- Oracle ASM (https://es.wikipedia.org/wiki/Automatic_Storage_Management)

Tech: Tablespaces (2/3)

```
dbnasr5022:/ORA/dbs13/ACCLOG 26T    25T  772G  98% /ORA/dbs13/ACCLOG_RAC50
dbnasr5032:/ORA/dbs08/ACCLOG 25T    23T  2.9T  89% /ORA/dbs08/ACCLOG_RAC50
dbnasr5032:/ORA/dbs11/ACCLOG        25T   22T  3.9T  85% /ORA/dbs11/ACCLOG_RAC50
dbnasr5032:/ORA/dbs06/ACCLOG        32T   31T  1.1T  97% /ORA/dbs06/ACCLOG_RAC50
dbnasr5022:/ORA/dbs05/ACCLOG        34T   33T  1.7T  96% /ORA/dbs05/ACCLOG_RAC50
dbnasr5031:/ORA/dbs02/ACCLOG        14T   11T  2.8T  80% /ORA/dbs02/ACCLOG_RAC50
dbnasr5032:/ORA/dbs12/ACCLOG        25T   21T  4.2T  84% /ORA/dbs12/ACCLOG_RAC50
dbnasr5011:/ORA/dbs0a/ACCLOG        35G   24G   12G  67% /ORA/dbs0a/ACCLOG_RAC50
dbnasr5022:/ORA/dbs07/ACCLOG        29T   27T  2.2T  93% /ORA/dbs07/ACCLOG_RAC50
dbnasr5011:/ORA/dbs00/ACCLOG       216G  2.6G  214G   2% /ORA/dbs00/ACCLOG_RAC50
dbnasr5032:/ORA/dbs10/ACCLOG        25T   24T  1.3T  95% /ORA/dbs10/ACCLOG_RAC50
dbnasr5022:/ORA/dbs09/ACCLOG        25T   23T  3.0T  89% /ORA/dbs09/ACCLOG_RAC50
dbnass:/CRS/dbs03/ACCLOG           2.0G  4.0M  2.0G   1% /CRS/dbs03/ACCLOG
dbnasr5071:/ORA/dbs03/ACCLOG       640G  400G  241G  63% /ORA/dbs03/ACCLOG_RAC50
dbnasr5021:/ORA/dbs04/ACCLOG       248G  205G   44G  83% /ORA/dbs04/ACCLOG_RAC50
dbnasr5032:/ORA/dbs14/ACCLOG        25T   23T  2.7T  90% /ORA/dbs14/ACCLOG_RAC50
dbnass:/CRS/dbs00/ACCLOG           5.0G  3.9G  1.2G  78% /CRS/dbs00/ACCLOG
dbnass:/CRS/dbs02/ACCLOG           2.0G  4.2M  2.0G   1% /CRS/dbs02/ACCLOG
dbnasr5022:/ORA/dbs15/ACCLOG        63T   28T   36T  44% /ORA/dbs15/ACCLOG_RAC50
dbnasr5032:/ORA/dbs16/ACCLOG        69T   30T   40T  43% /ORA/dbs16/ACCLOG_RAC50
dbnasr5022:/ORA/dbs17/ACCLOG        30T   29T  1.1T  97% /ORA/dbs17/ACCLOG_RAC50
```

Tech: Tablespaces (3/3)

```
db-dbnasb401:/backup/dbs05/ACCLOG   48T   47T  1.3T  98% /backup/dbs05/ACCLOG
db-dbnasb402:/backup/dbs04/ACCLOG  101T   91T  9.4T  91% /backup/dbs04/ACCLOG
db-dbnasb402:/backup/dbs08/ACCLOG   80T   74T  6.4T  93% /backup/dbs08/ACCLOG
dbnasc501-c:/acclog_backup00 200T  200T  422G 100% /backup/dbs00/ACCLOG
db-dbnasb401:/backup/dbs01/ACCLOG   60T   57T  3.2T  95% /backup/dbs01/ACCLOG
db-dbnasb401:/backup/dbs07/ACCLOG   91T   83T  7.3T  92% /backup/dbs07/ACCLOG
db-dbnasb401:/backup/dbs03/ACCLOG  101T   94T  6.7T  94% /backup/dbs03/ACCLOG
db-dbnasb402:/backup/dbs02/ACCLOG   60T   57T  3.1T  95% /backup/dbs02/ACCLOG
db-dbnasb402:/backup/dbs06/ACCLOG   48T   47T  1.3T  98% /backup/dbs06/ACCLOG
```

Tech: BRIN indexes

PostgreSQL BRIN Indexes (https://www.postgresql.org/docs/current/static/brin-intro.html)

BRIN stands for Block Range Index. BRIN is designed for handling very large tables in which certain columns have some natural correlation with their physical location within the table.

A block range is a group of pages that are physically adjacent in the table; for each block range, some summary info is stored by the index.

Oracle Exadata Storage Indexes (http://www.oracle.com/technetwork/testcontent/o31exadata-354069.html)

Replication (1/3)

One or more copies (replicas/slaves) of part (or the full) primary/master database systems

Most systems implement it by having DML (Data Manipulation Language) operations appended to a log (Binary Log in MySQL, Write Ahead Log in PostgreSQL, Undo/Redo log in Oracle) which is later copied and re-applied in the replicas

Can be **asynchronous** (most typical) or **synchronous**, *physical* or *logical*.

Replicas are normally configured as **Read Only**.

Used for HA solutions, Disaster Recovery or Load scaling

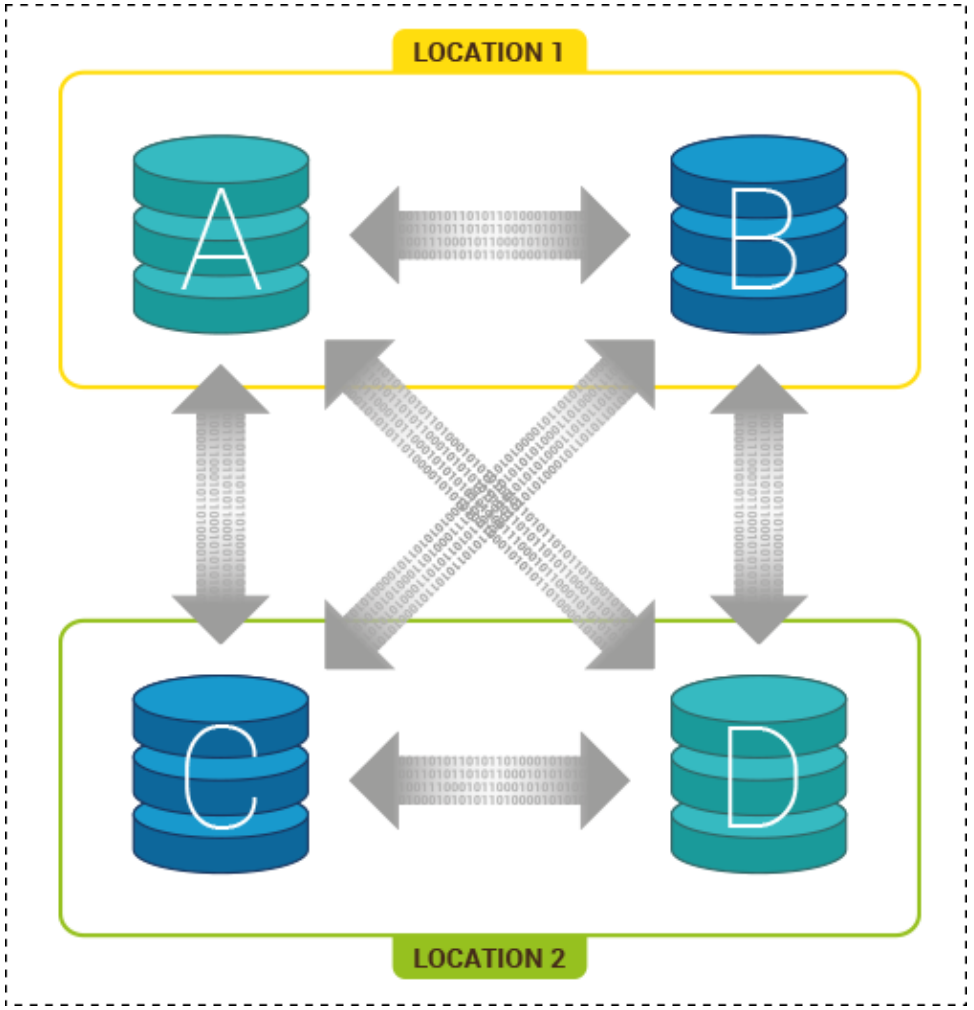
Replication (2/3)

Booking.Com Scaling (https://blog.booking.com/mysql_slave_scaling_and_more.html)

```
-----
| A |
-----
  |
  +------+------+------------+
  |      |      |            |
  V      V      V            V
-----  -----  -----        -----
| B |  | C |  | D |        | E |
-----  -----  -----        -----
                             |
                             +------+------+
                             |      |      |
                             V      V      V
                           -----  -----  -----
                           | F |  | G |  | H |
                           -----  -----  -----
```

Replication: Multi-master (3/3)

PostgreSQL BDR (https://www.2ndquadrant.com/en/resources/bdr/)



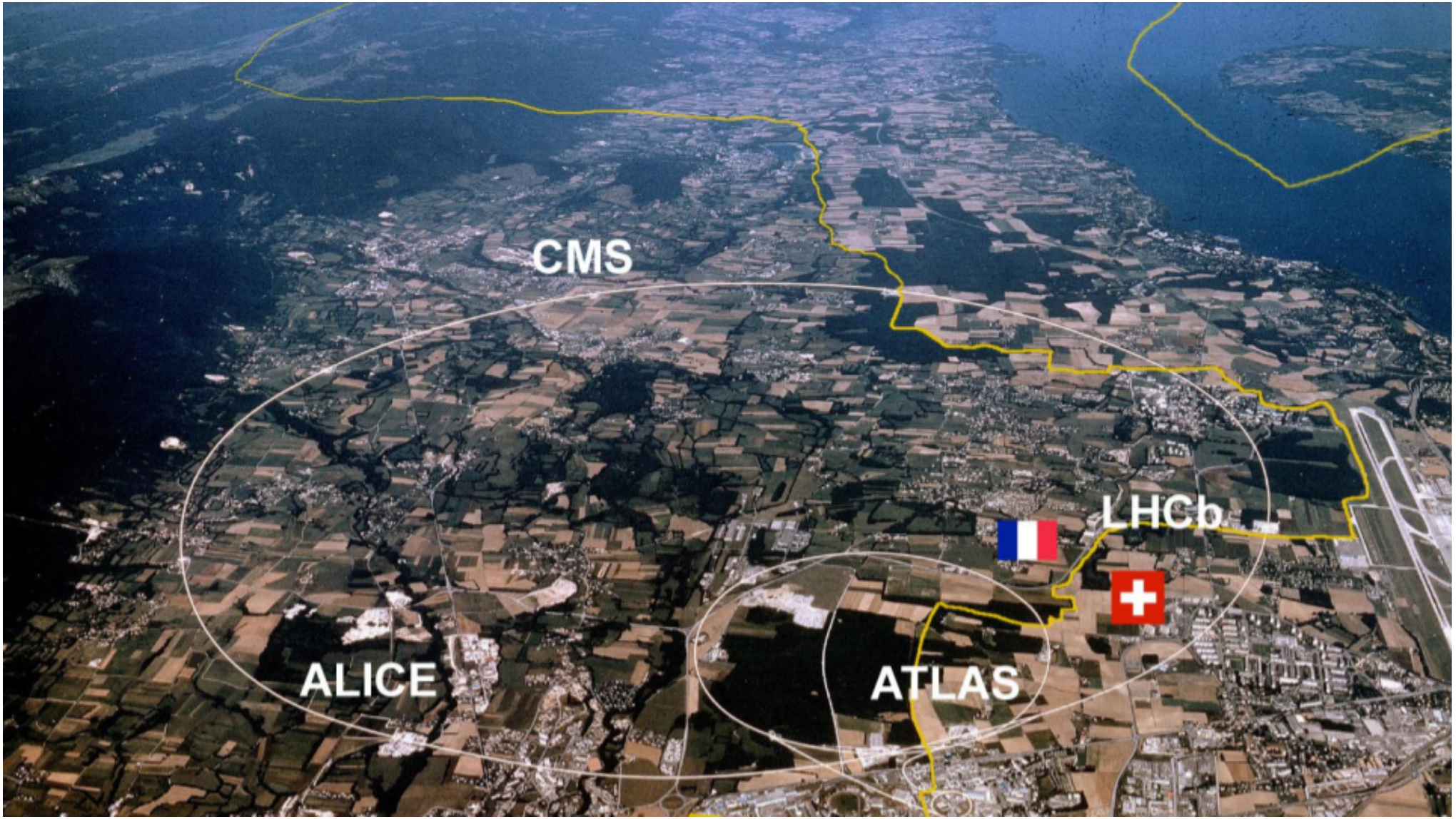
Sharding

Shared nothing. The data is partitioned in multiple systems and operations are parallelized among all the ndoes

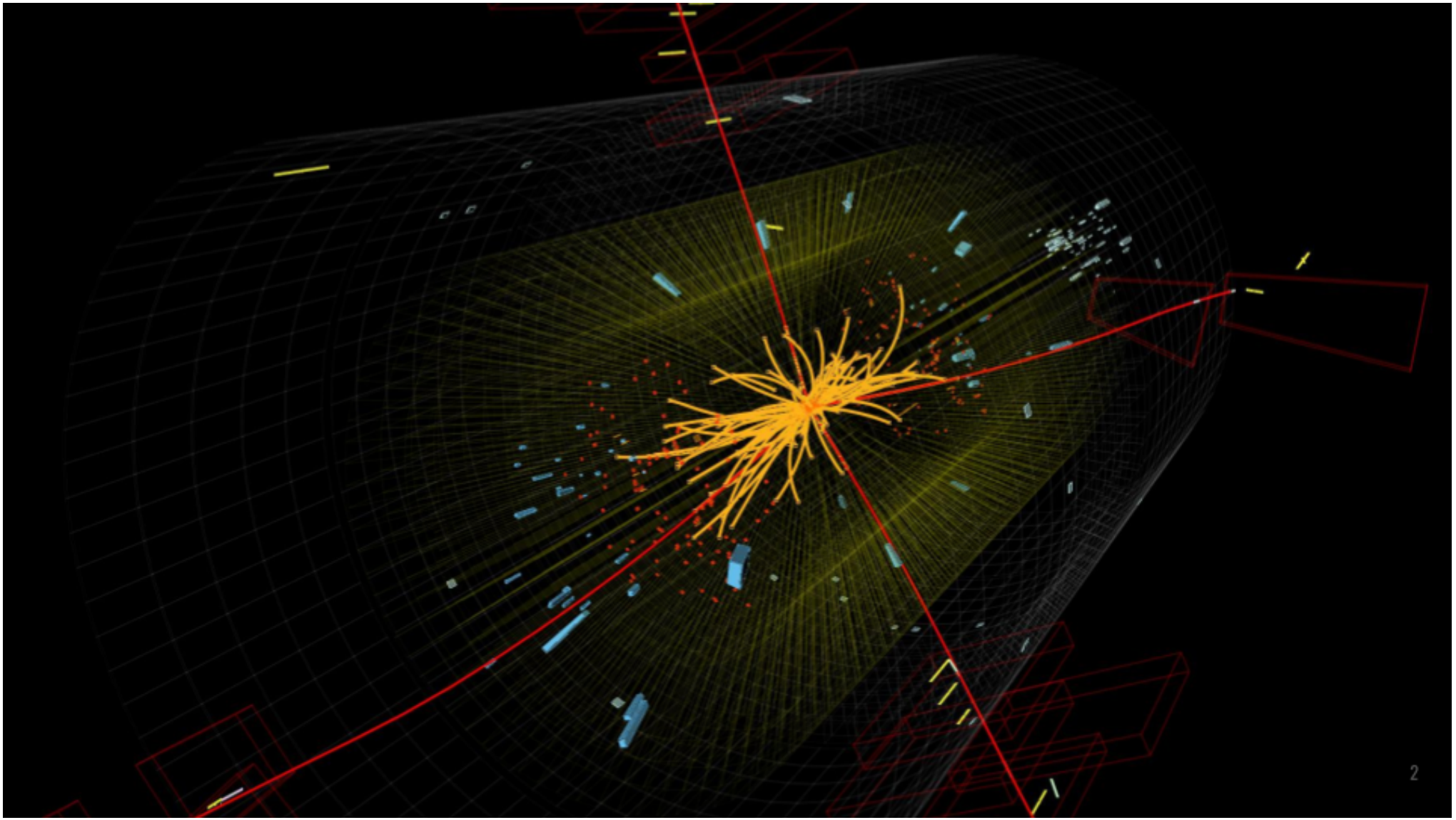
Sharding strategies can vary:

- Uniform distribution for load balancing
- By application logic: customer, provider, geographically, ...

CERN

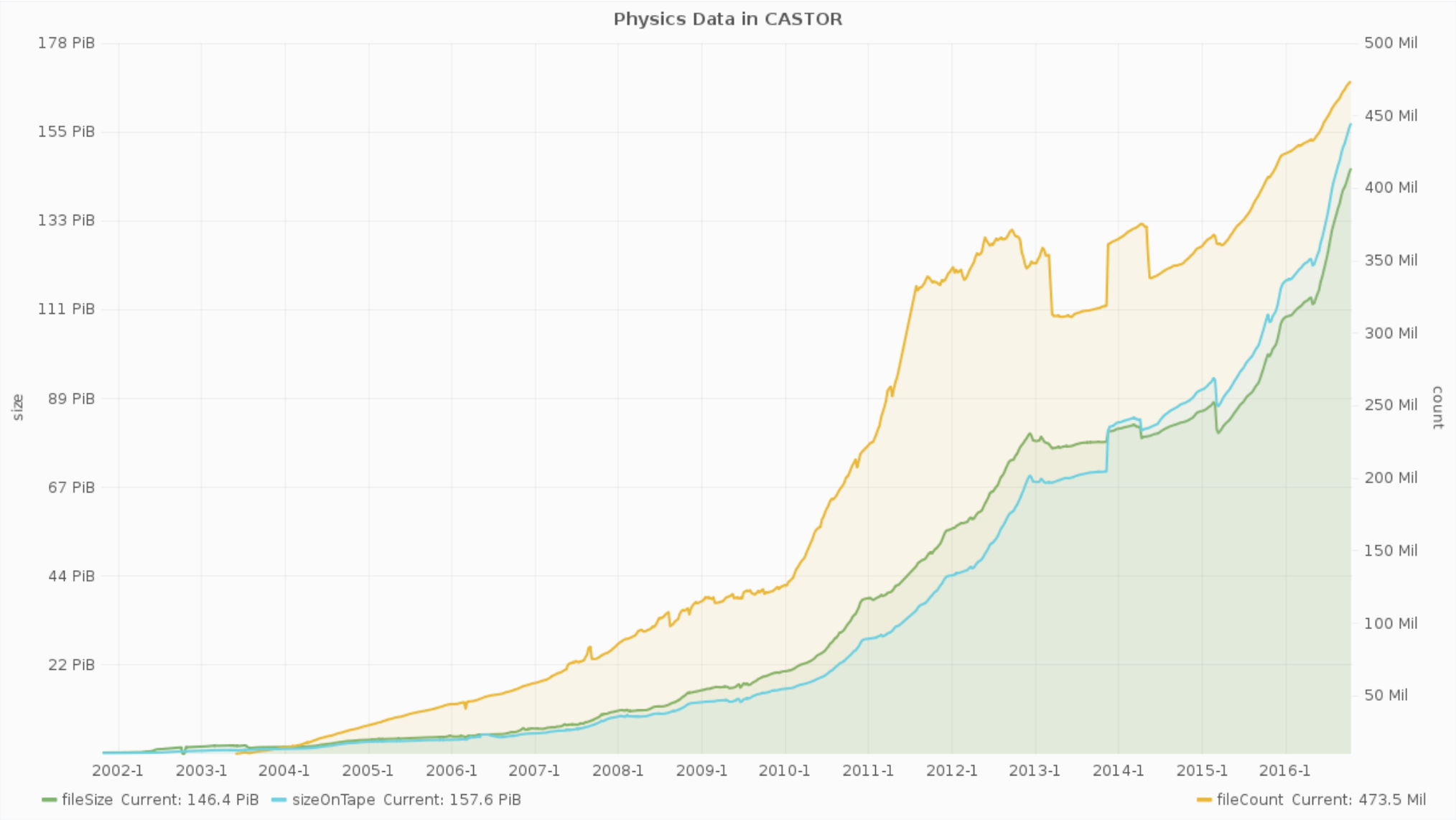


CERN

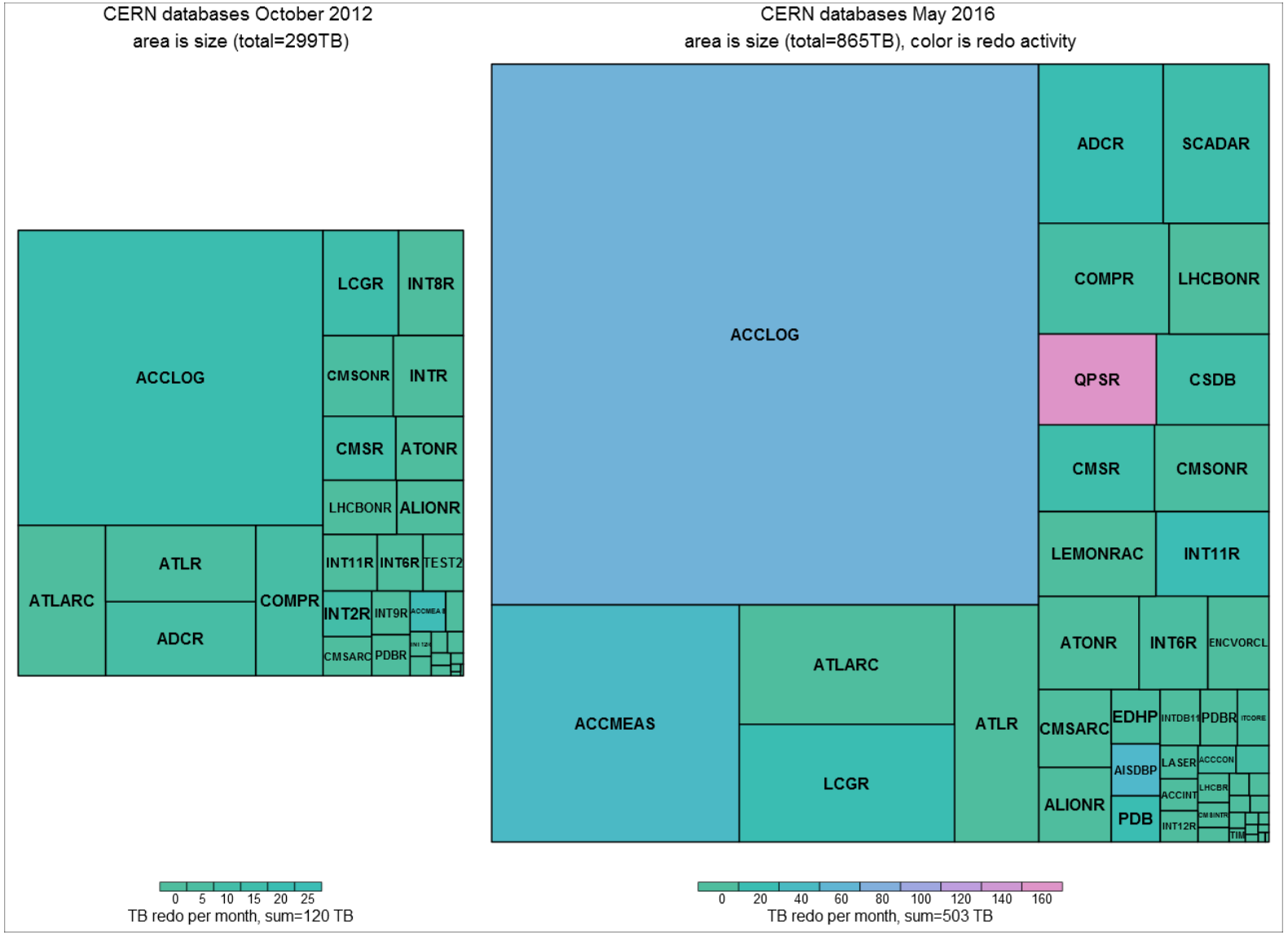


CERN

CERN: Raw Data



CERN: RDBMS


CERN databases October 2012
area is size (total=299TB)
ACCLOG
LCGR
INT8R
CMSONR
INTR
CMSR
ATONR
LHCBONR
ALIONR
ATLR
ATLARC
ADCR
COMPR
INT11R
INT6R
TEST2
INT2R
INT9R
CMSARC
PDBR
0 5 10 15 20 25
TB redo per month, sum=120 TB
CERN databases May 2016
area is size (total=865TB), color is redo activity
ACCLOG
ADCR
SCADAR
COMPR
LHCBONR
QPSR
CSDB
CMSR
CMSONR
LEMONRAC
INT11R
ATONR
INT6R
ENCVORCL
ACCMEAS
ATLARC
LCGR
ATLR
CMSARC
EDHP
PDBR
AISDBP
LASER
ACCCON
ACCINT
LHCBR
ALIONR
PDB
INT12R
0 20 40 60 80 100 120 140 160
TB redo per month, sum=503 TB

CERN: RDBMS

- ~70 Oracle RAC clusters with ~30 replicas
- ~390 MySQL (~10 replicas)
- ~125 PostgreSQL (~5 replicas)

```
MySQL and PostgreSQL databases are managed by the DB on Demand service, automatizing
database lifecycle and operations
```

CERN: NAS Overview

CPU utilization

Filer	Average CPU util [%]
dbnash5111	76%
dbnasdb1021	73%
dbnass5241	70%
dbnast1332	65%
dbnash5131	60%
dbnass5231	57%
dbnasdb1022	56%
dbnass5211	55%
dbnass5222	46%
dbnash5122	43%
dbnasb401	43%
dbnass5221	40%
dbnass5242	39%
dbnasda1011	38%
dbnasda1012	38%

1 2 3 4

Disk utilization

Metric	Aggregate disk busy [%]
dbnass5221	99%
dbnasdb1021	99%
dbnass5231	94%
dbnass5222	90%
dbnash5122	88%
dbnasdb1022	80%
dbnast1351	78%
dbnass5241	75%
dbnast1331	65%
dbnast1332	63%
dbnasda1011	61%
dbnash5131	56%
dbnash5121	51%
dbnass5242	47%
dbnass5232	46%

1 2 3 4

Read latency

Metric	Average NFS read latency
dbnasdb1021	93.20 ms
dbnasdb1022	77.38 ms
dbnash5122	58.72 ms
dbnass5231	32.71 ms
dbnass5222	28.05 ms
dbnass5221	20.24 ms
dbnasda1011	19.31 ms
dbnass5241	14.60 ms
dbnast1351	12.48 ms
dbnasr5022	7.53 ms
dbnasr5032	5.11 ms
dbnasb402	5.02 ms
dbnast1331	4.59 ms
dbnasb401	4.42 ms
dbnash5112	4.06 ms

1 2 3 4

Write latency

Metric	Average NFS read latency
dbnasdb1022	15.87 ms
dbnasdb1021	11.52 ms
dbnass5221	6.41 ms
dbnass5231	5.40 ms
dbnass5241	4.92 ms
dbnass5222	4.36 ms
dbnass5232	2.21 ms
dbnash5122	1.33 ms
dbnasp01	1.31 ms
dbnasr5031	1.22 ms
dbnash5131	1.18 ms
dbnast1331	1.10 ms
dbnass5242	0.98 ms
dbnass5212	0.98 ms
dbnasda1011	0.96 ms

1 2 3

Throughput

9.31 GBs

IOps

379.64K iops

Total raw capacity

5.08 PB

Used raw capacity

3.92 PB

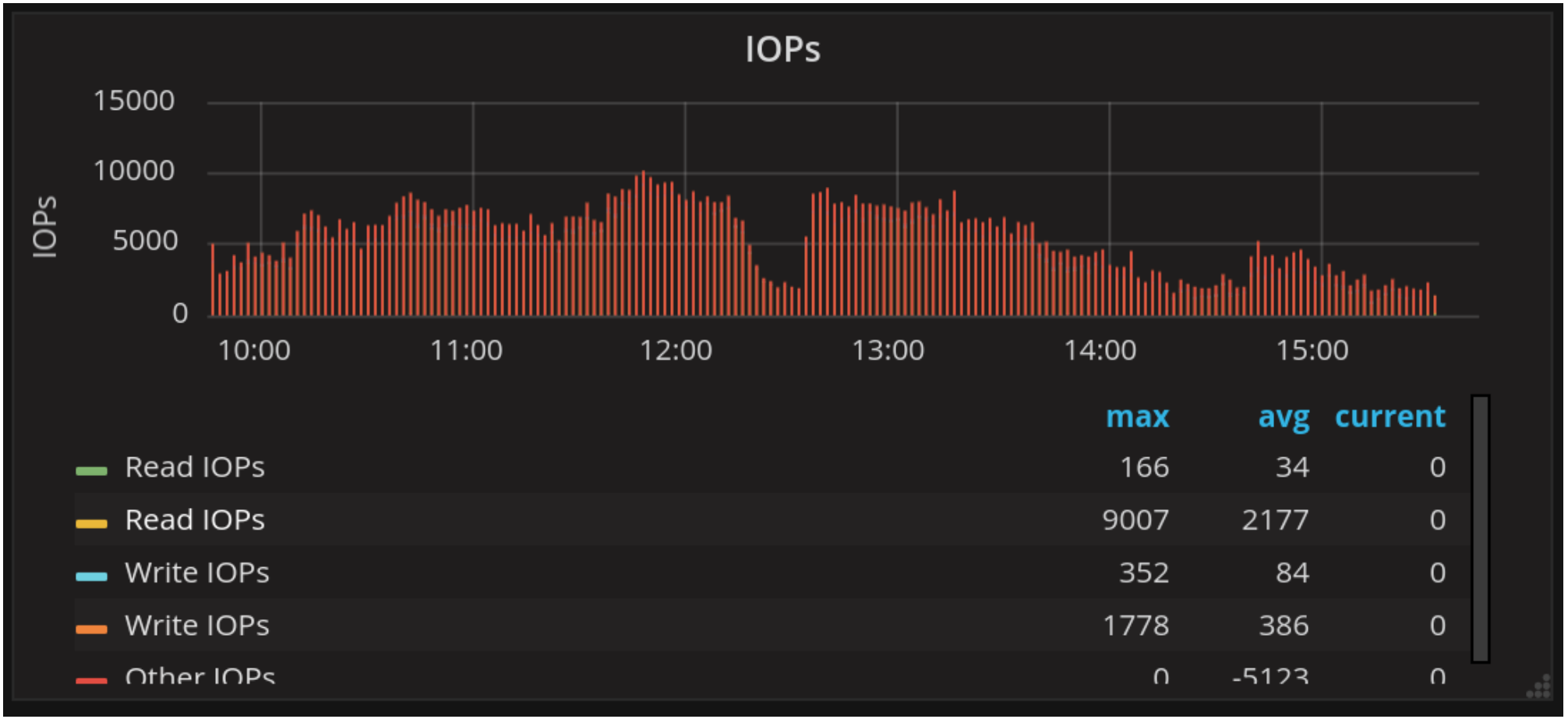
Saved by efficiency

3.45 PB

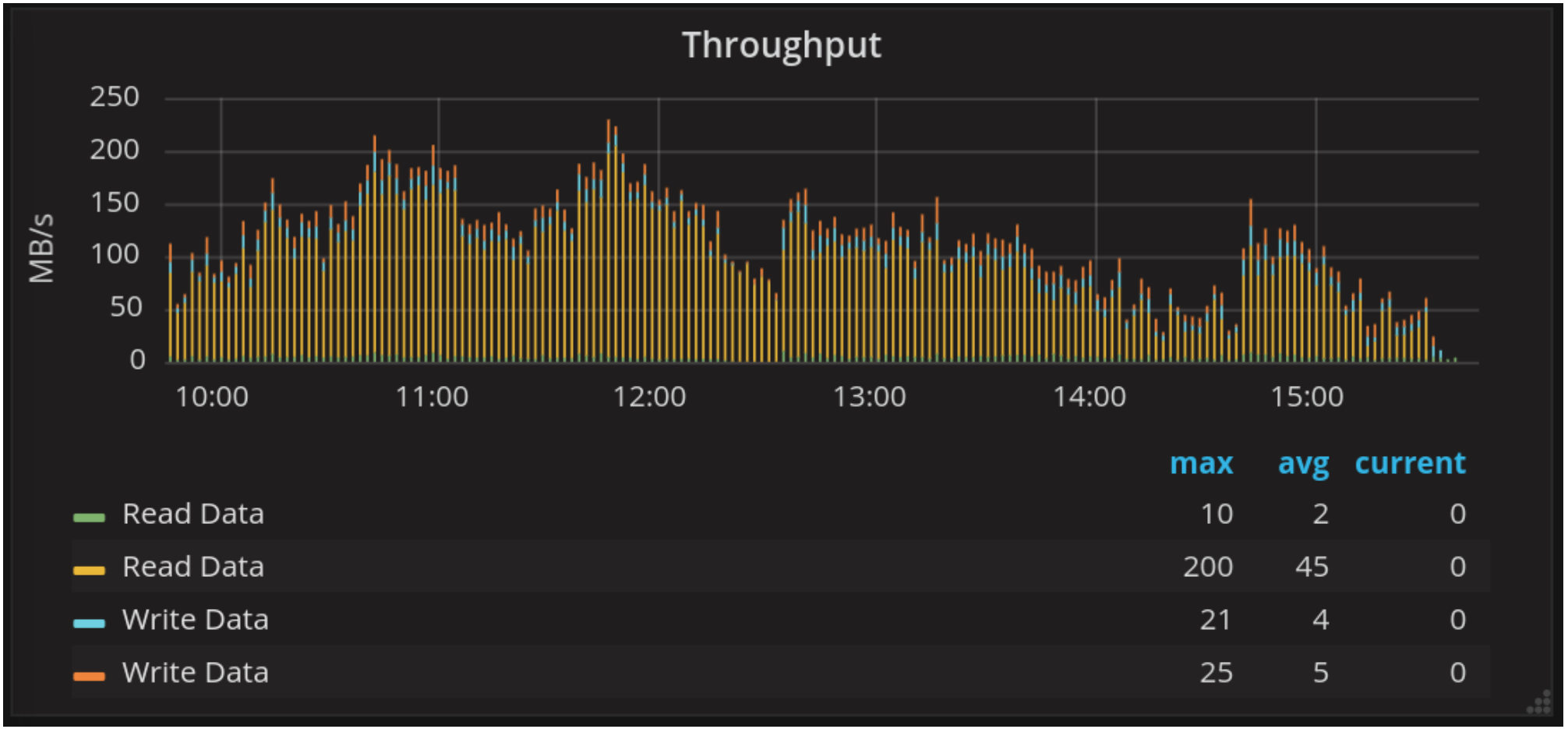
Total data stored

7.36 PB

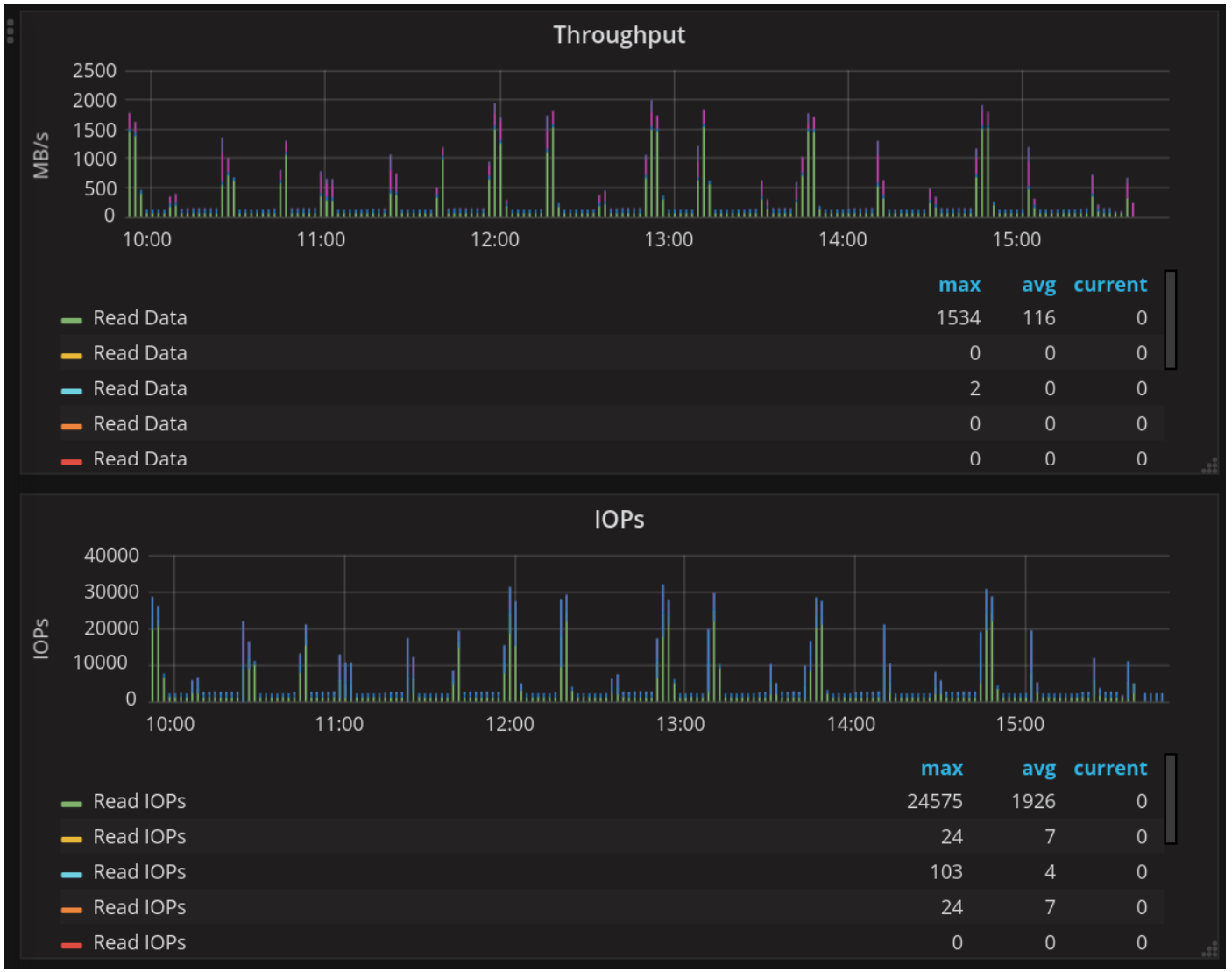
NAS IOPs


IOPs
15000
10000
5000
0
IOPs
10:00
11:00
12:00
13:00
14:00
15:00
max avg current
Read IOPs 166 34 0
Read IOPs 9007 2177 0
Write IOPs 352 84 0
Write IOPs 1778 386 0
Other IOPs 0 -5123 0

NAS Throughput



NAS QPSR


Throughput
MB/s
2500
2000
1500
1000
500
0
10:00
11:00
12:00
13:00
14:00
15:00
max avg current
Read Data 1534 116 0
Read Data 0 0 0
Read Data 2 0 0
Read Data 0 0 0
Read Data 0 0 0
IOPs
IOPs
40000
30000
20000
10000
0
10:00
11:00
12:00
13:00
14:00
15:00
max avg current
Read IOPs 24575 1926 0
Read IOPs 24 7 0
Read IOPs 103 4 0
Read IOPs 24 7 0
Read IOPs 0 0 0

Thank you

Ignacio Coterillo
Computer Engineer, CERN
ignacio.coterillo.coz@cern.ch (mailto:ignacio.coterillo.coz@cern.ch)