## StashCache benchmarking on the IGWN infrastructure a programmatic approach

XCache DevOps Meeting 18<sup>th</sup> February 2021





Gabriele Gaetano Fronzé for the INFN-Torino Computing Group

## Summary

- IGWN Data distribution
- IGWN Data checker: a benchmarking tool

# IGWN Data Distribution

XCache DevOps Meeting 18th February 2021 Gabriele Gaetano Fronzé for the INFN-Torino Computing Group

53

6

S

6

#### Base bricks

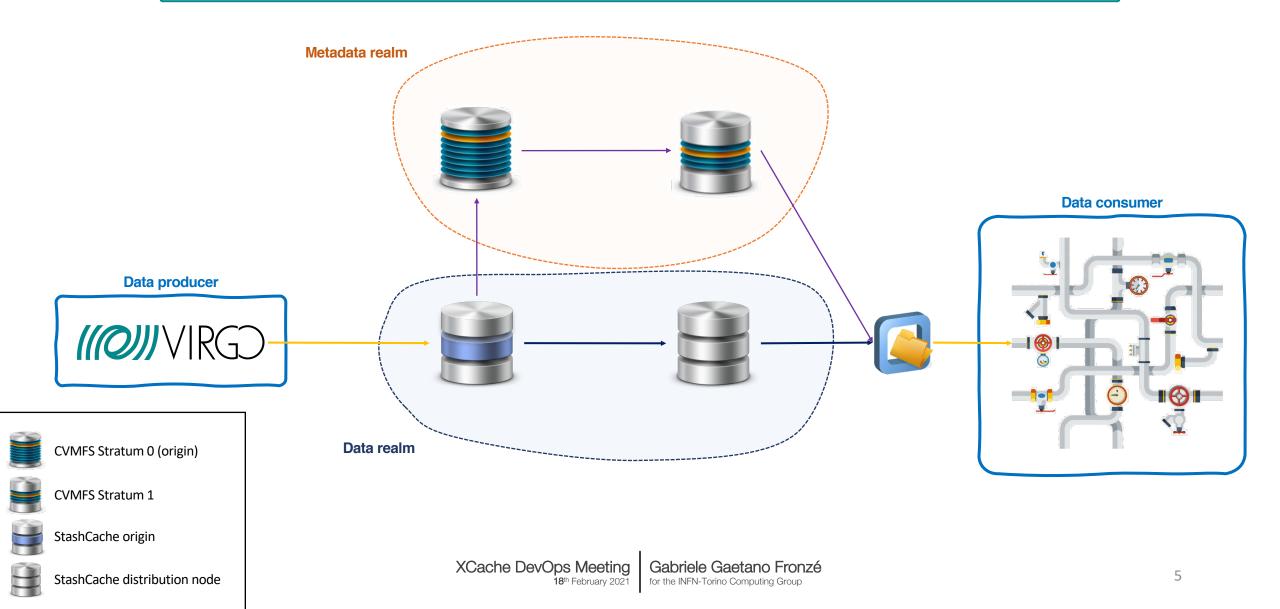


#### CVMFS = Data catalog

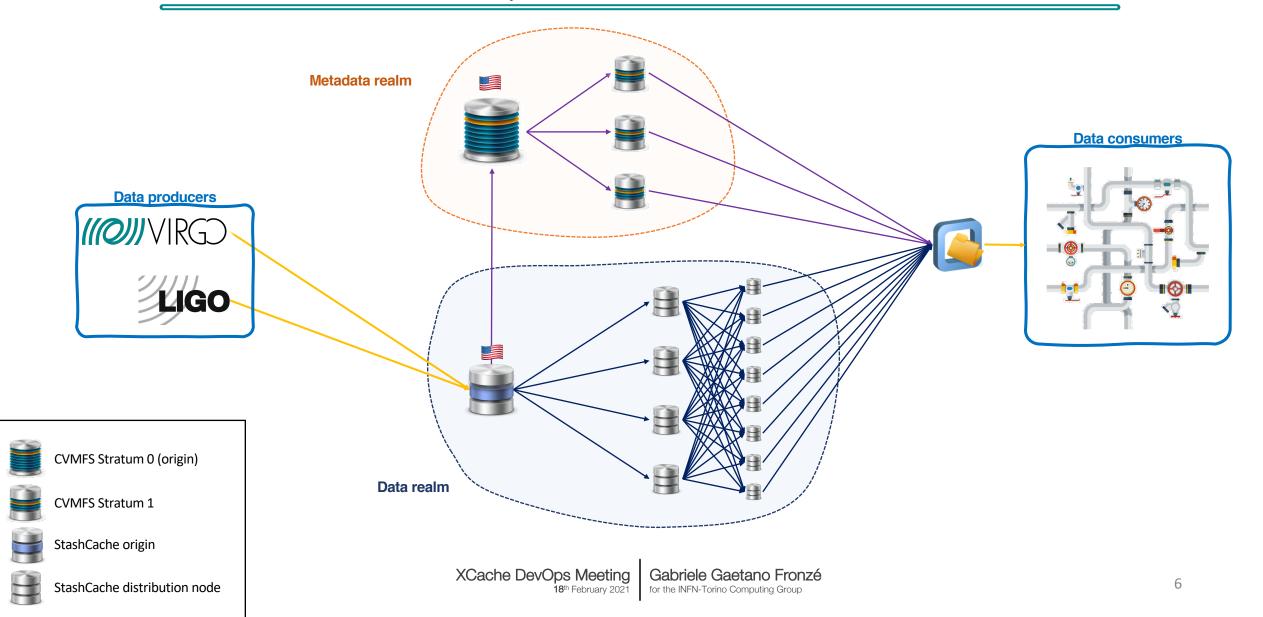


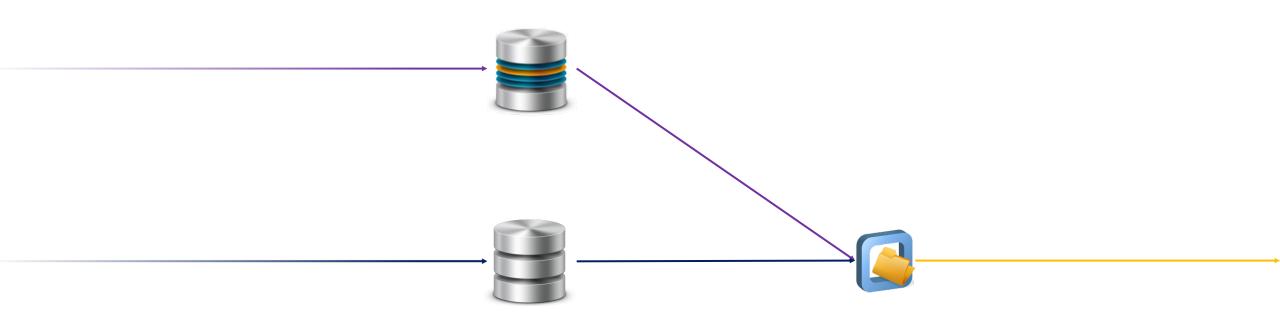
#### StashCache = Data storage

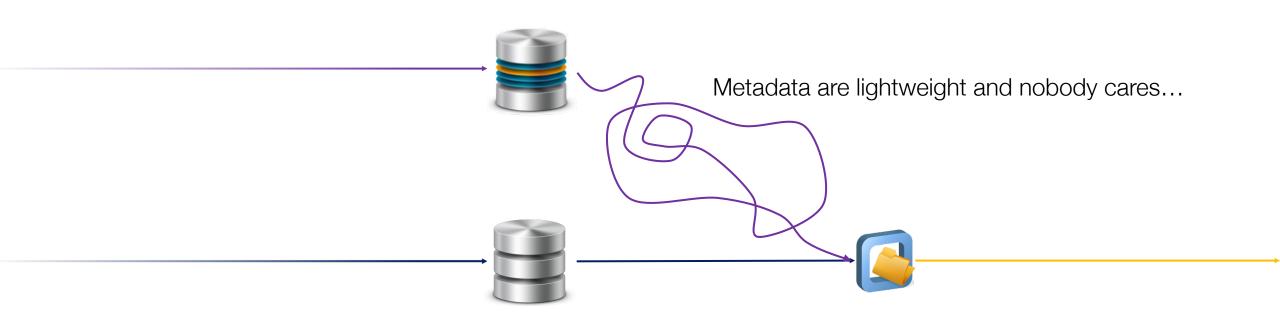
#### Base framework

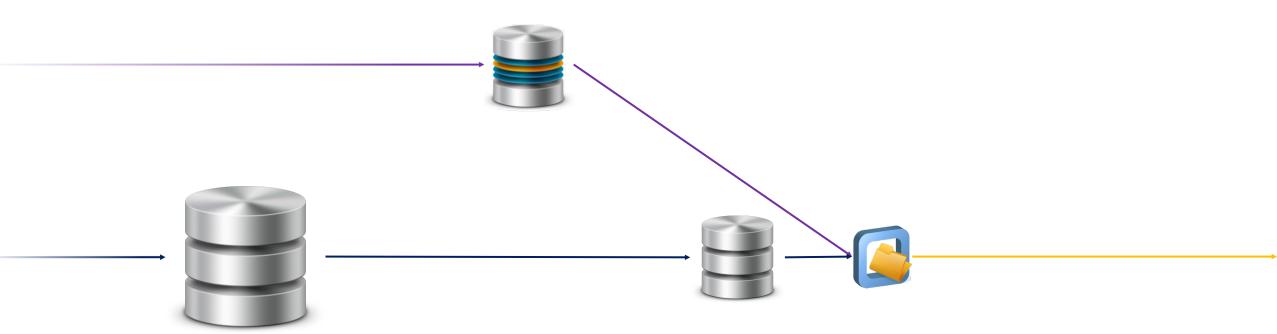


## Complete framework

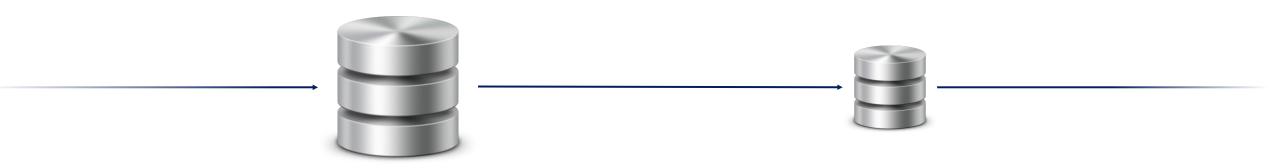








Data are heavy! Keep it as close as possible!



#### Computing centers installed or are installing local caches to improve data access

# IGWN Data Checker

A data benchmarking tool

 
 XCache DevOps Meeting 18<sup>th</sup> February 2021
 Gabriele Gaetano Fronzé for the INFN-Torino Computing Group

## A tool for "last mile" measurements

#### A simple tool to:

- measure data access times
- measure file corruption probability
- asses local caches status

#### from the IGWN computing facilities



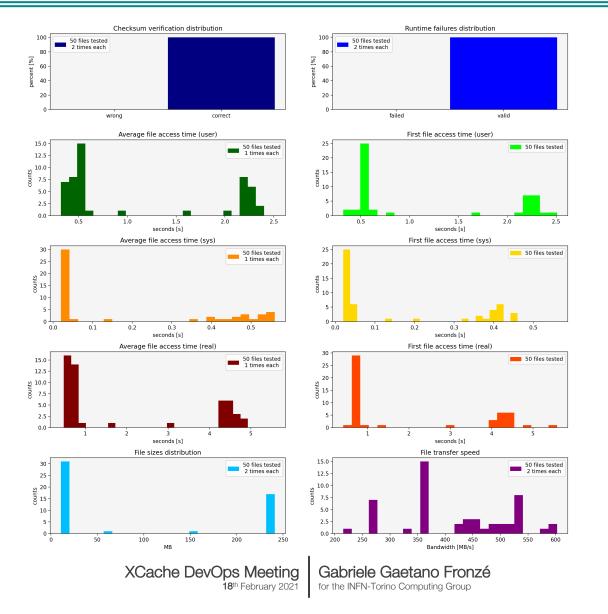
#### A tool for "last mile" measurements



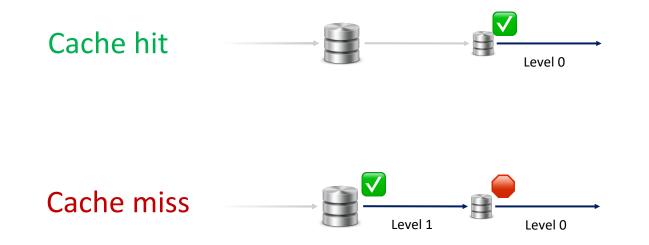
## Example output

Checksum verification rate	Runtime failures rate
Average access user time	First access user time
Average access system time	First access system time
Average access real time	First access real time
File size distribution	Storage bandwidth

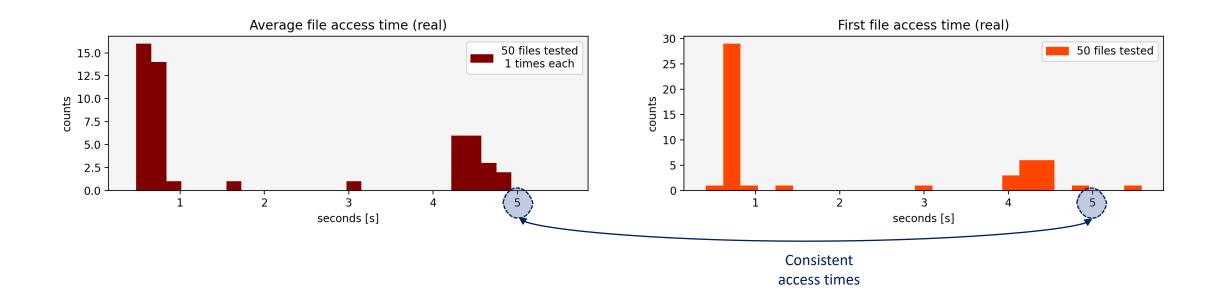
### Example output



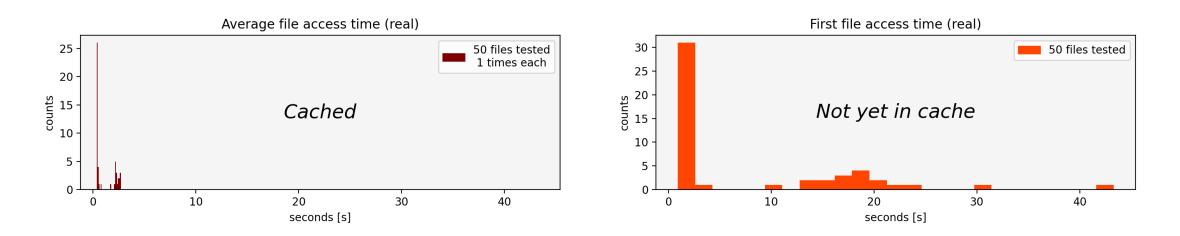
#### When a cache is in place two things can happen when one tries to access a file:



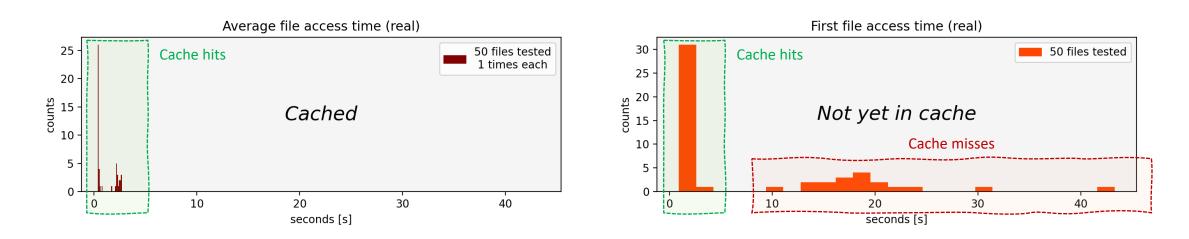
If data is already present in the nearest cache, the first access time should be consistent with the average of the following ones.



Otherwise, the first access should result in a cache miss, triggering the cache to populate that entry. Following accesses will find the file in cache and be much faster!

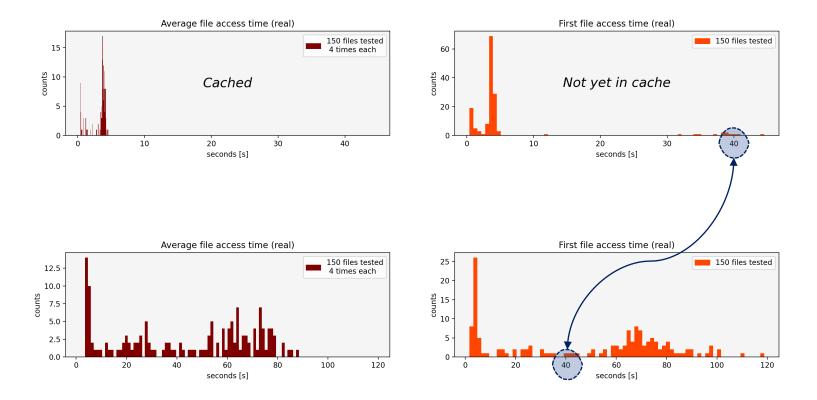


Otherwise, the first access should result in a cache miss, triggering the cache to populate that entry. Following accesses will find the file in cache and be much faster!



### Underperforming cache detection

Comparing multiple local caches one can highlight problematic ones.



#### Both computing centers have a local cache.

The first one behaves as expected, some files are not yet cached but most of them results in a cache hit.

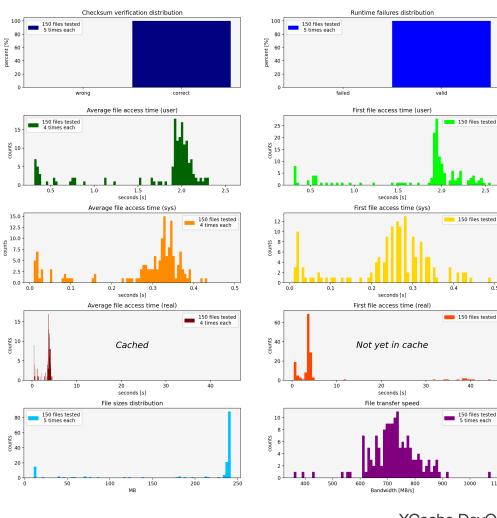
Following accesses are faster.

The second computing center shows slow access times (x3 wrt cache misses in the first) and following acceses are almost identical.

There is something wrong with the second CC cache to be addressed!

## Preliminary @ CNAF and PIC

#### Input file: output-CNAF-LONG.json



#### 100 50 files tested 100 50 files tested 2 times each 2 times each 60 2 40 wrong correct failed valid Average file access time (user) First file access time (user) 50 files tested 1 times each 50 files tested 25 20 20 <sup>원</sup> 15 5 15 10 0.50 0.75 1.00 1.25 1.50 1.75 2.00 2.25 0.25 0.50 0.75 1.00 1.25 1.50 1.75 2.00 2.25 0.25 seconds [s] seconds [s] Average file access time (sys) First file access time (sys) 50 files tested 50 files tested \_ 1 times each 25 . 20 Ē 15 10 01 0.00 0.05 0.10 0.15 0.20 0.25 0.30 0.35 0.00 0.05 0.10 0.15 0.20 0.25 0.30 seconds [s] seconds [s] Average file access time (real) First file access time (real) 50 files tested 30 50 files tested 2 1 times each 25 20 ₽ 15 Cached Not yet in cache ' 10 10 10 20 40 10 20 seconds [s] seconds [s] File sizes distribution File transfer speed 50 files tested 50 files tested 2 times each 2 times each n 20 2 1

500

400

600 700 800

Bandwidth [MB/s]

900

#### Input file: output-PIC.json

Runtime failures distribution

Checksum verification distribution

50

100

150

200

250

XCache DevOps Meeting Gabriele Gaetano Fronzé 18th February 2021 for the INFN-Torino Computing Group

1100

0.5

1000

1100

0.35



# This tool can be packed in a suite of jobs to be sent around periodically to health-check our data distribution network and detect failures and other issues

# Thank you for your attention!

