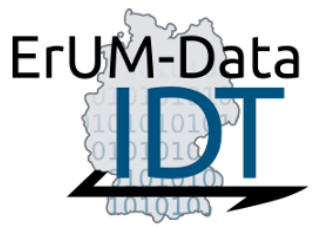
Xcache status

Nikolai Hartmann

LMU Munich

February 24, 2020



What is XCache?

- Disk caching proxy using xrootd (`libXrdFileCache.so`)

  → What is xrootd?
  - Remote access ~~protocol~~ software framework
  - ... with many features (proxy, cluster, caching, storage system, third-party-copy, authentication protocols, ...)
  - becoming the standard remote reading protocol at ATLAS/CMS

- Data is cached in blocks

- Simply prepend xcache server url - e.g.

  ```
  TFile::Open("root:[xcache-server]:[port]//[xrootd-path]")
  ```

- Optionally use rucio DIDs via N2N plugin:
  https://github.com/wyang007/rucioN2N-for-Xcache
  → allows usage of rucio DIDs instead of xrootd path
  → tracks identical files distributed at different locations
  (internal symlink `.../scope/XX/YY/filename`)

What is XCache?

- Disk caching proxy using xrootd (`libXrdFileCache.so`)

  → What

  - Rem
  - ... w
    third
  - beco

  software framework

  proxy, cluster, caching, storage system,

  ication protocols, ...)

  emote reading protocol at ATLAS/CMS

- Data is cached in blocks

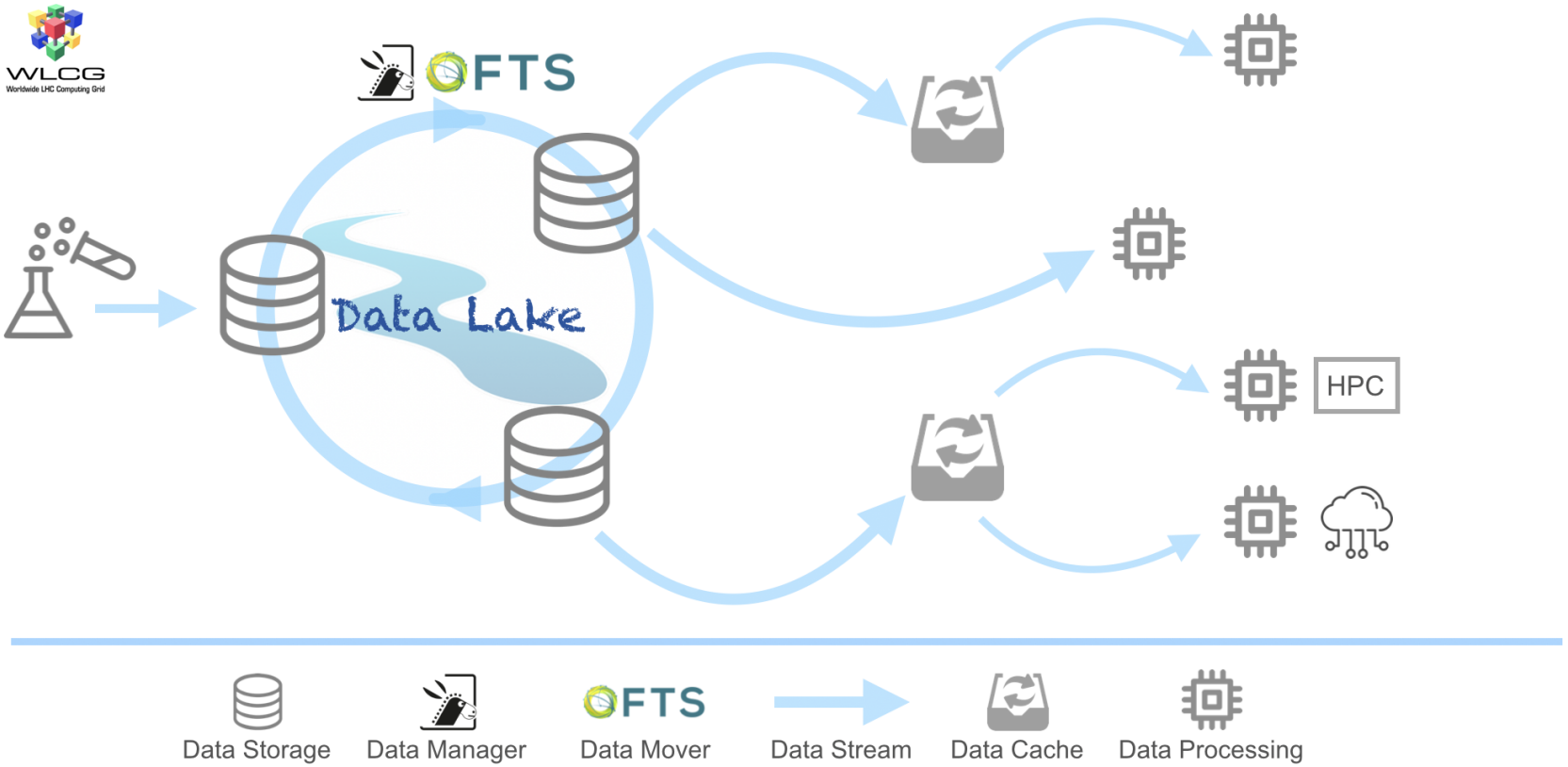
- Simply prepend xcache server url - e.g.

```
TFile::Open("root:[xcache-server]:[port]//[xrootd-path]")
```

- Optionally use rucio DIDs via N2N plugin:
  https://github.com/wyang007/rucioN2N-for-Xcache
  → allows usage of rucio DIDs instead of xrootd path
  → tracks identical files distributed at different locations
  (internal symlink `.../scope/XX/YY/filename`)

Why?

- There is a push towards a storage model around "data lakes"
  $\rightarrow$ sites with large storage
- Small storage sites should read remotely from closest data lake
  $\rightarrow$ hoped to reduce both hardware resources and manpower needs
- Caching might help for 2 things
  - Hide latency
    $\rightarrow$ push the problem of efficient remote reading (e.g. reading in sufficiently large blocks, parallel) to the cache server
  - Reduce WAN traffic
    $\rightarrow$ files that are accessed again might still be in cache



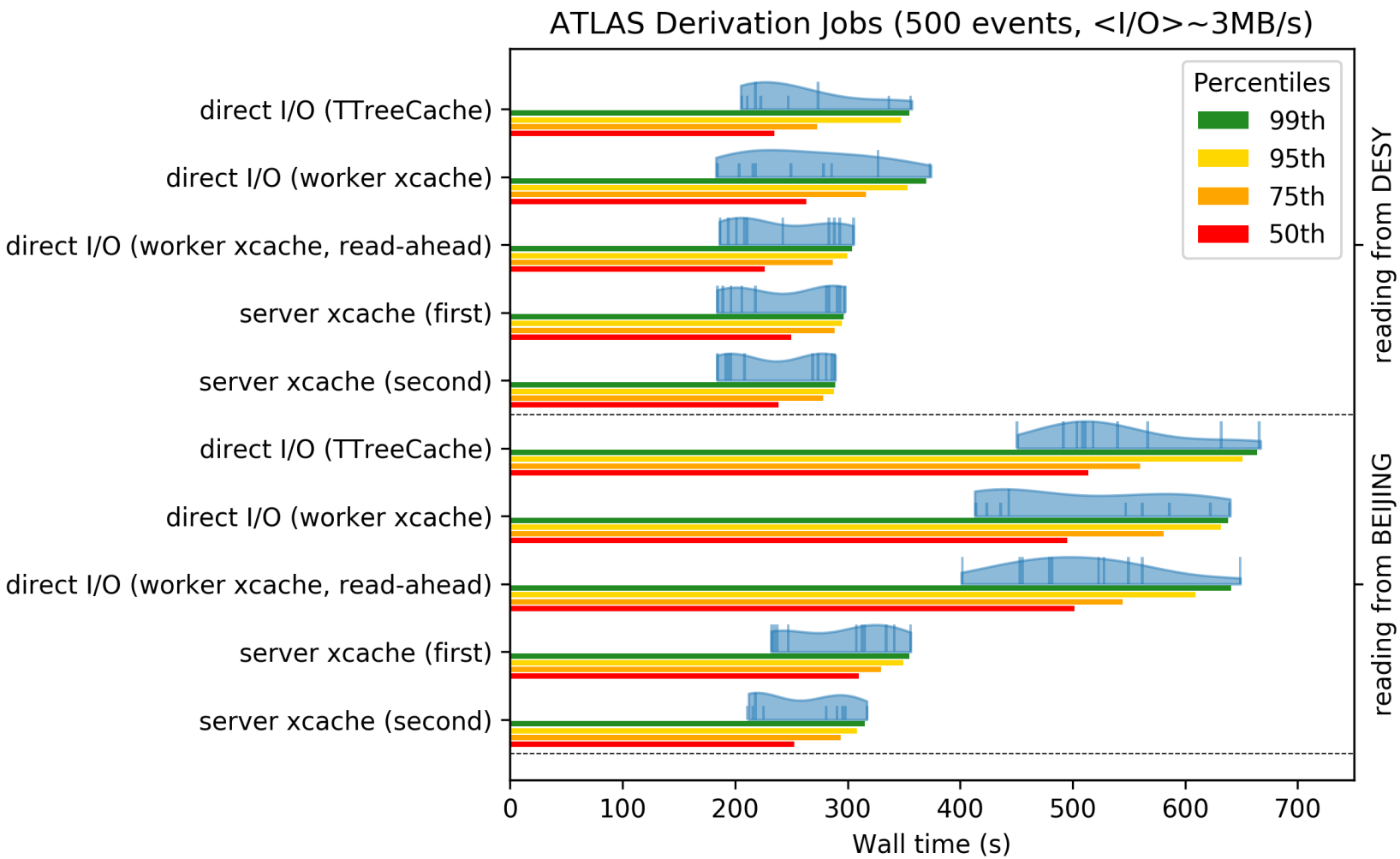
Datalakes, latency hiding and caching - Xavier Espinal (CERN)

Setup

- Hardware: Old dCache pool node (from 2012):
  - Dell R710, 2x6 core Xeon L5640, 32 GB RAM, 10 Gb Ethernet
  - 60 TB Raid-6 (2x12x3TB HDD)
    (switched to individual disks instead of RAID6)
- Xrootd version 4.10.0
- Setup w/ singularity SL6 image. Full configuration:
  https://gitlab.physik.uni-muenchen.de/Nikolai.Hartmann/xcache-singularity-lrz/
- XCache settings:

```
pfc.ram 14g
pfc.blocksize 1M
pfc.prefetch 10
```

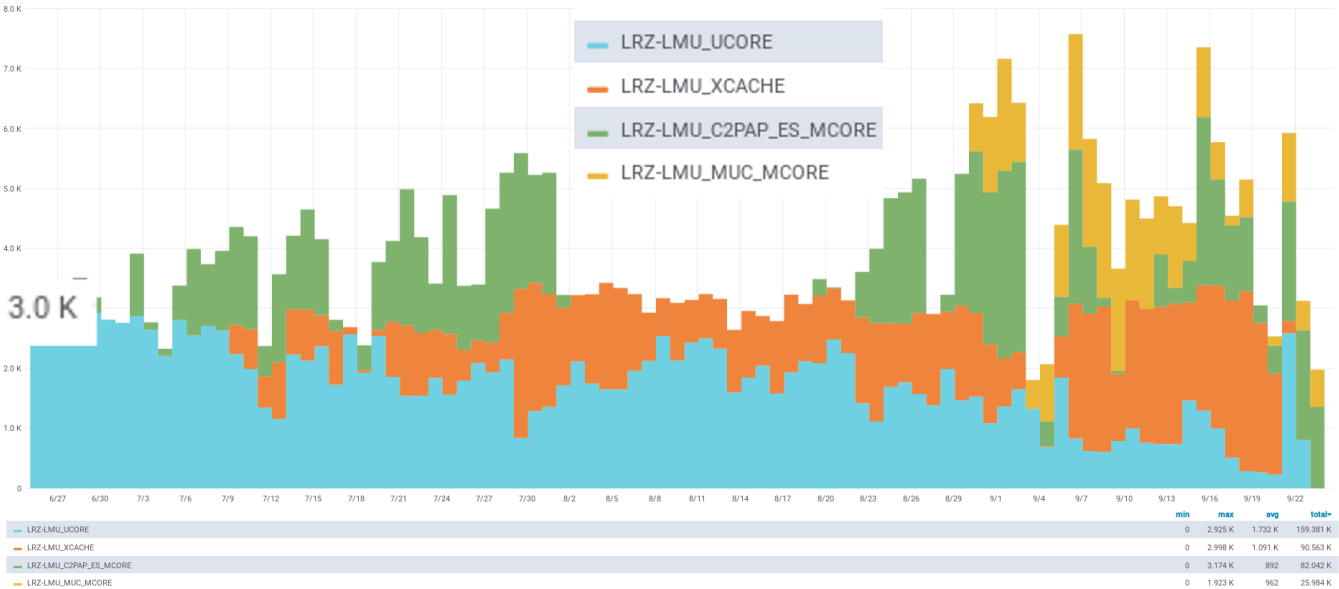
Initial studies on latency hiding



$\rightarrow$ caching on server hides latency better than caching on worker nodes

Test XCache in ATLAS production queue

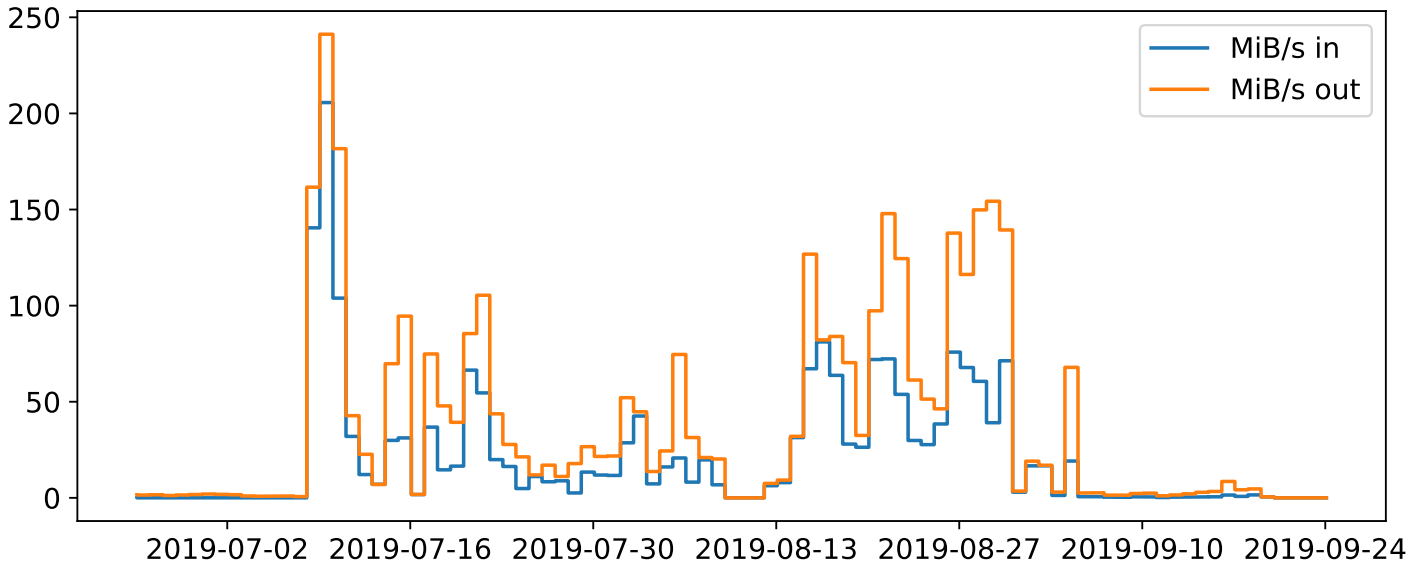
Running slots (jobs * CPUs) July - September 2019:



ATLAS production queue in Munich that retrieves all files via XCache

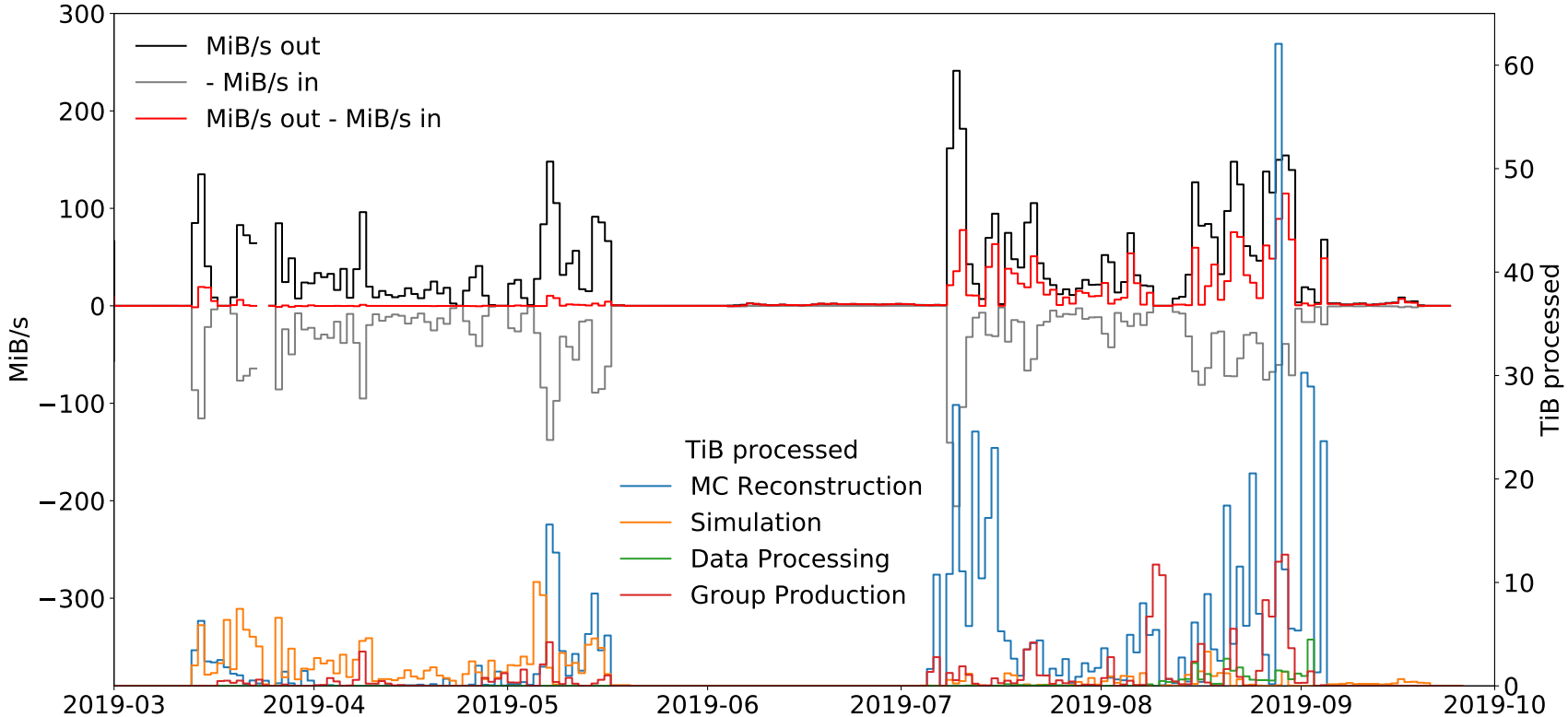
- Remote destination is nearby MPP Munich storage
- Can take a quite significant fraction of the jobs
- Works surprisingly well, given that all traffic goes through a single server

Caching works



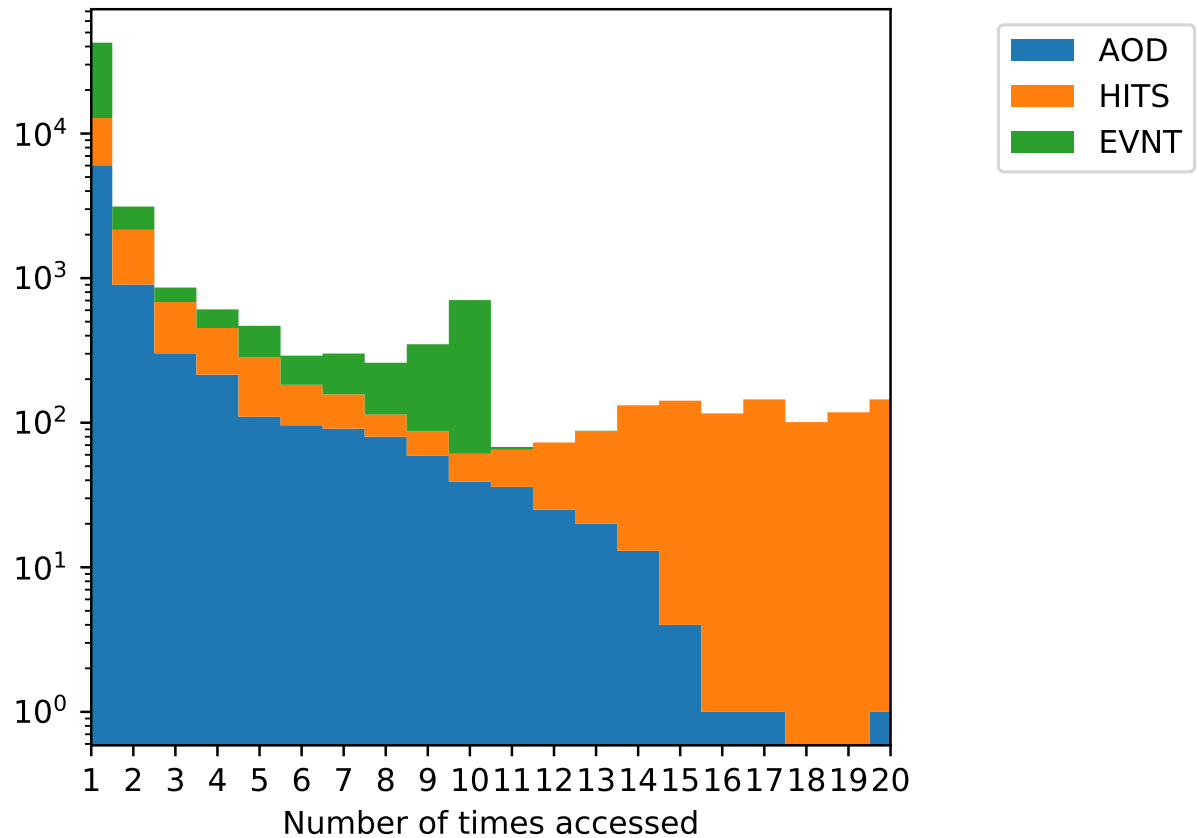
$\rightarrow$ Output volume already larger than input volume (≈ 1.8)

But hit rate depends on type of job



→ largest hit rate for MC Reconstruction (here mainly pileup overlay)

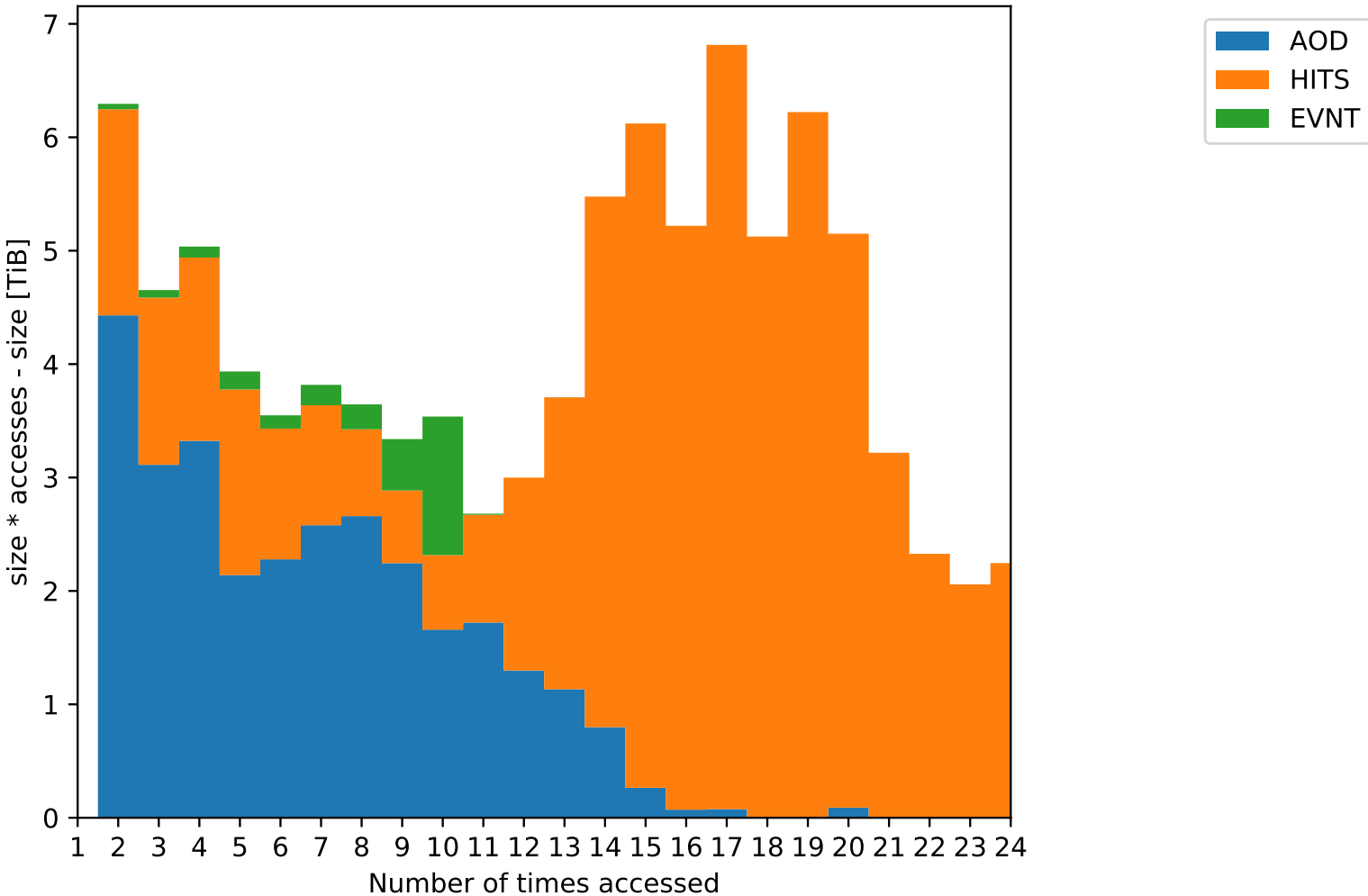
Access statistics from cinfo files



- Most reused files are HITS (pileup)
- EVNT files get reused when one file is processed via multiple jobs
- AOD files get reused for DAOD production (?)

Weighted by `size * accesses - size`

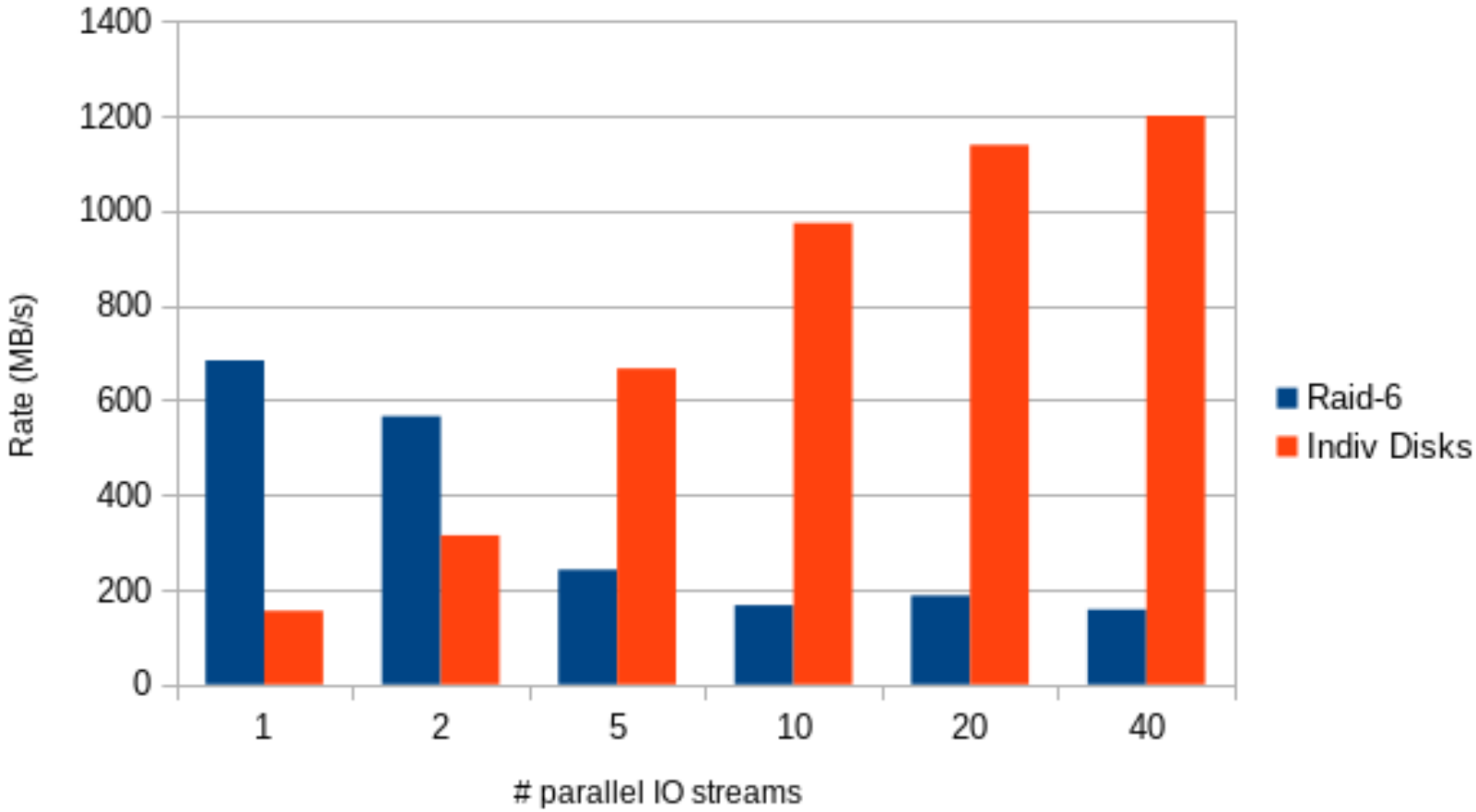
Corresponding reduction in WAN traffic
(w.r.t reading everything from remote without cache)



Performance for parallel reads - Raid6 vs single disks

Feedback from xrootd developers: Use multidisk-mode instead of Raid
(see slides from Matevž at XRootD workshop)

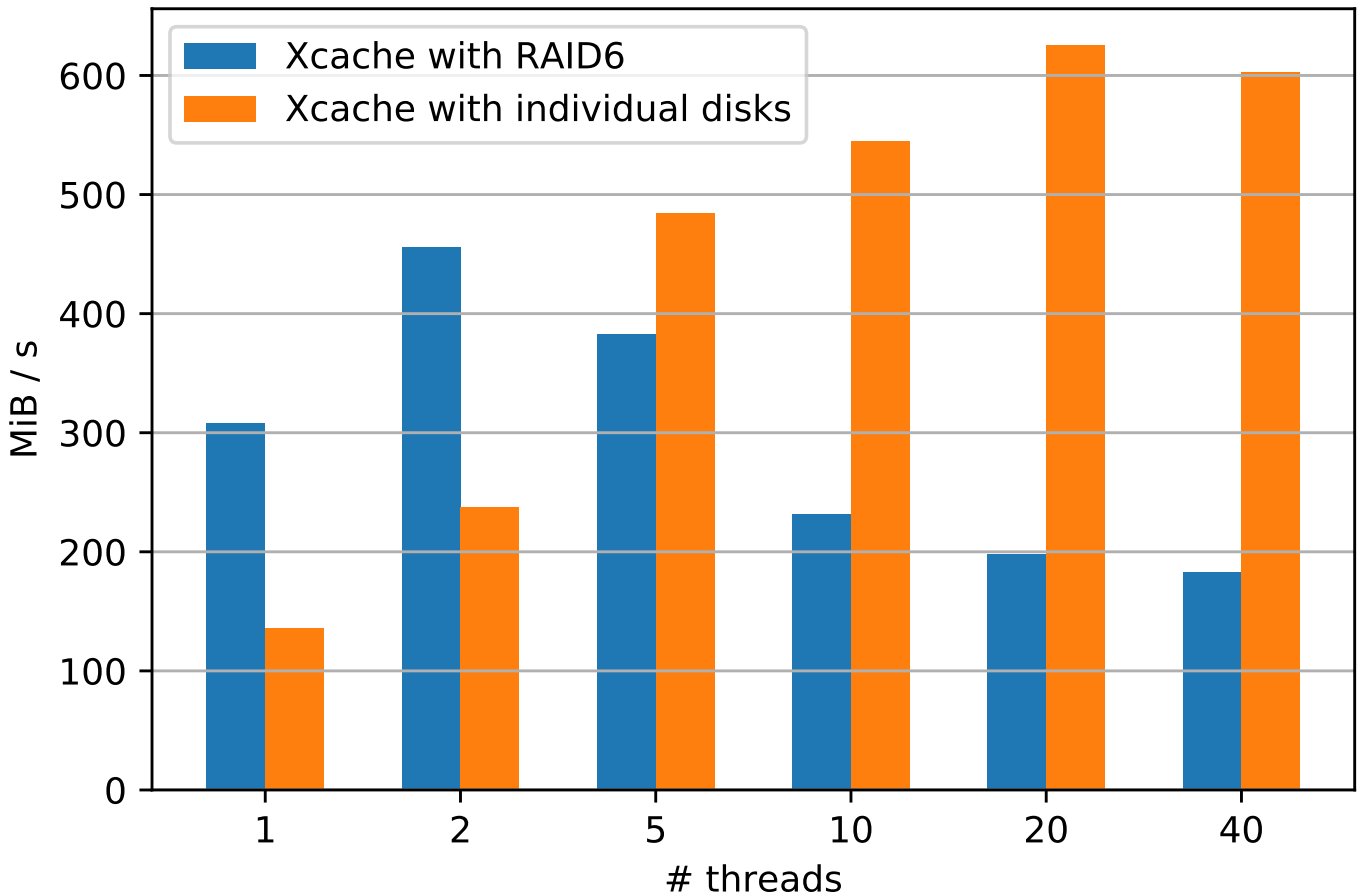
Raw reading tests at LRZ:



→ multi-disk mode might perform better than Raid for caching system

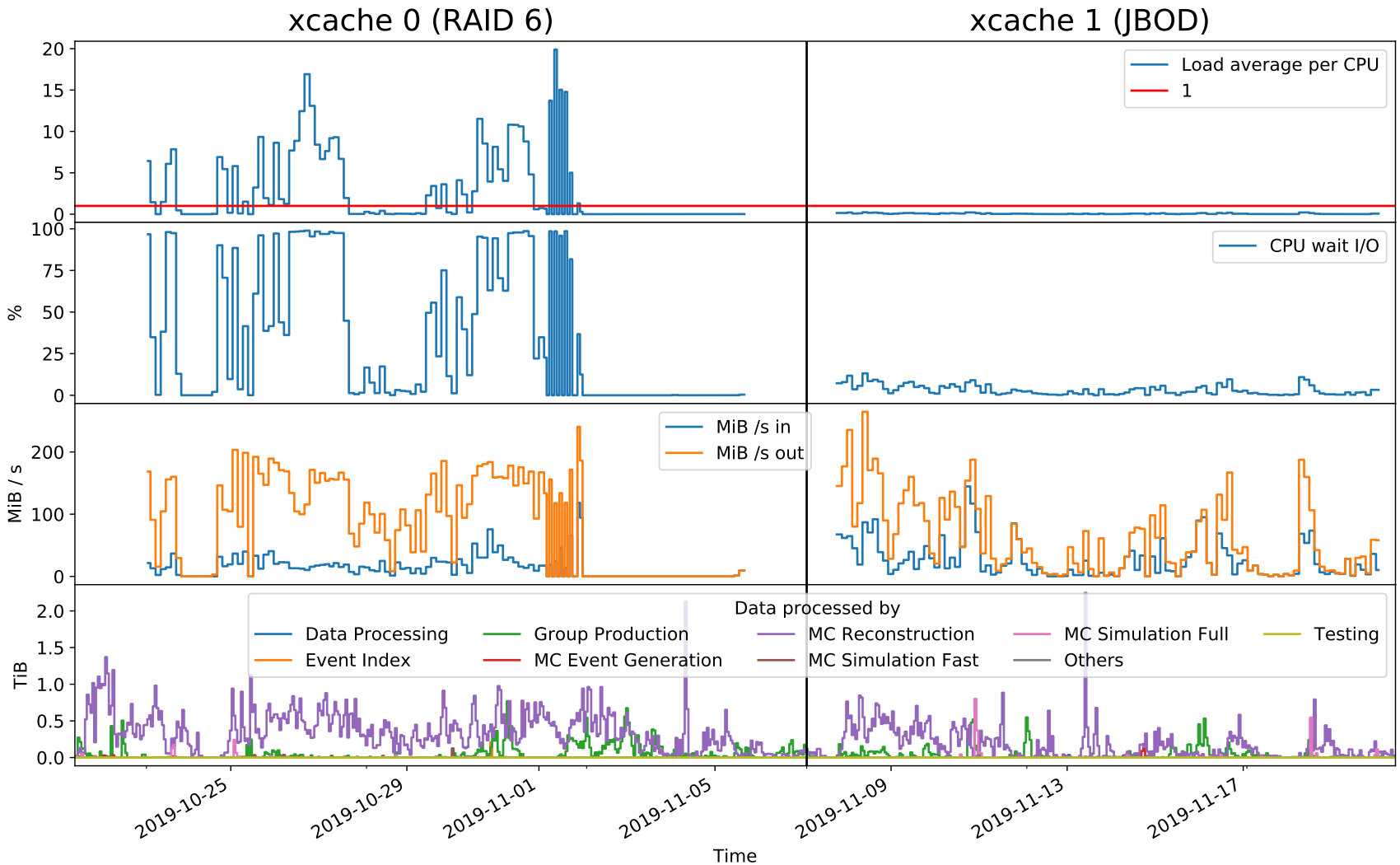
Performance for parallel reads - Raid6 vs single disks

Now similar test with an actual xcache setup:
(read random cached files through xcache, read from server)



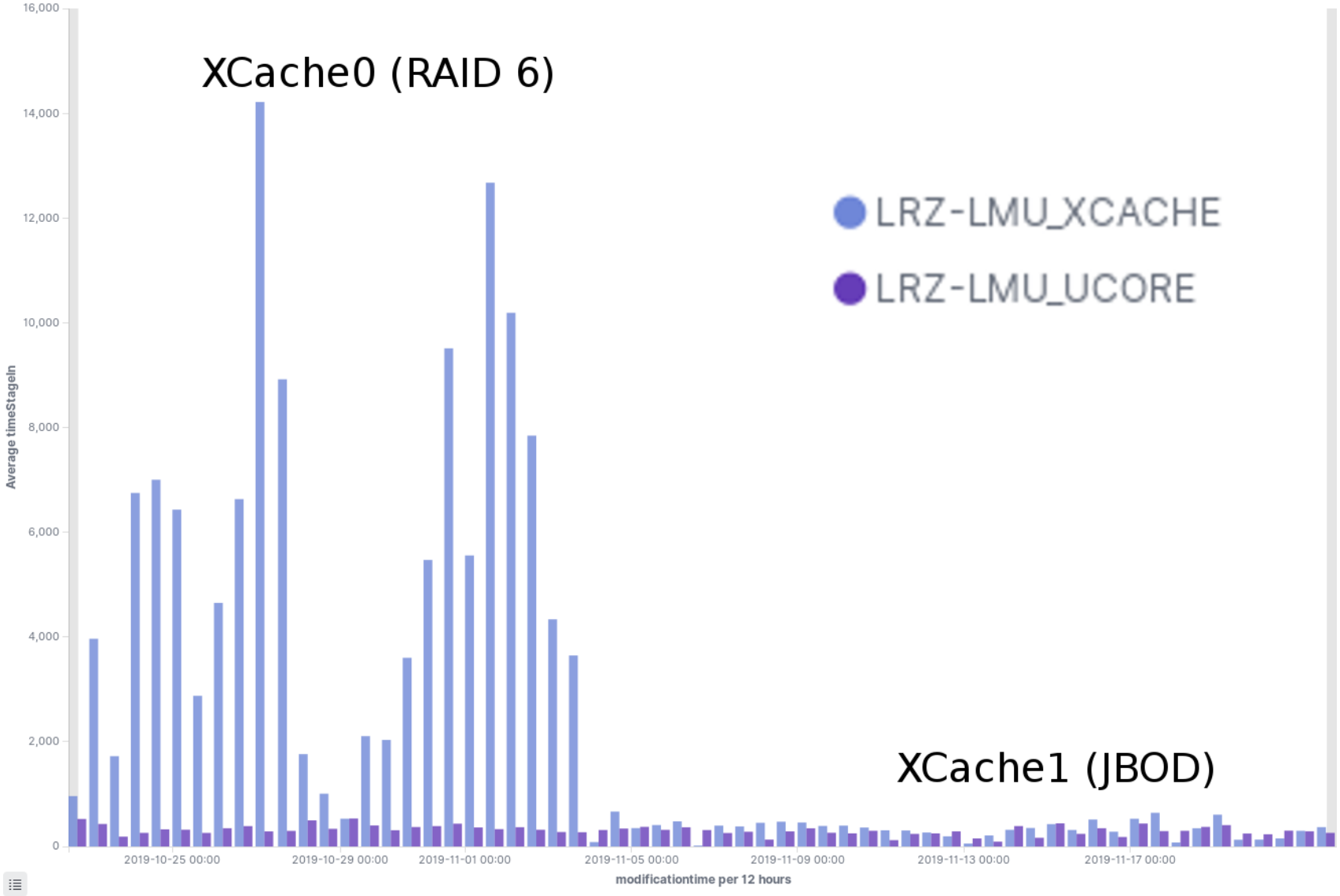
→ same conclusion - individual disks outperform RAID for parallel reads

Multidisk XCache in ATLAS production queue



→ load and wait CPU drastically reduced for multidisk mode setup!

Stage-in times



→ comparable stage-in times (with JBOD) as for non-xcache queue

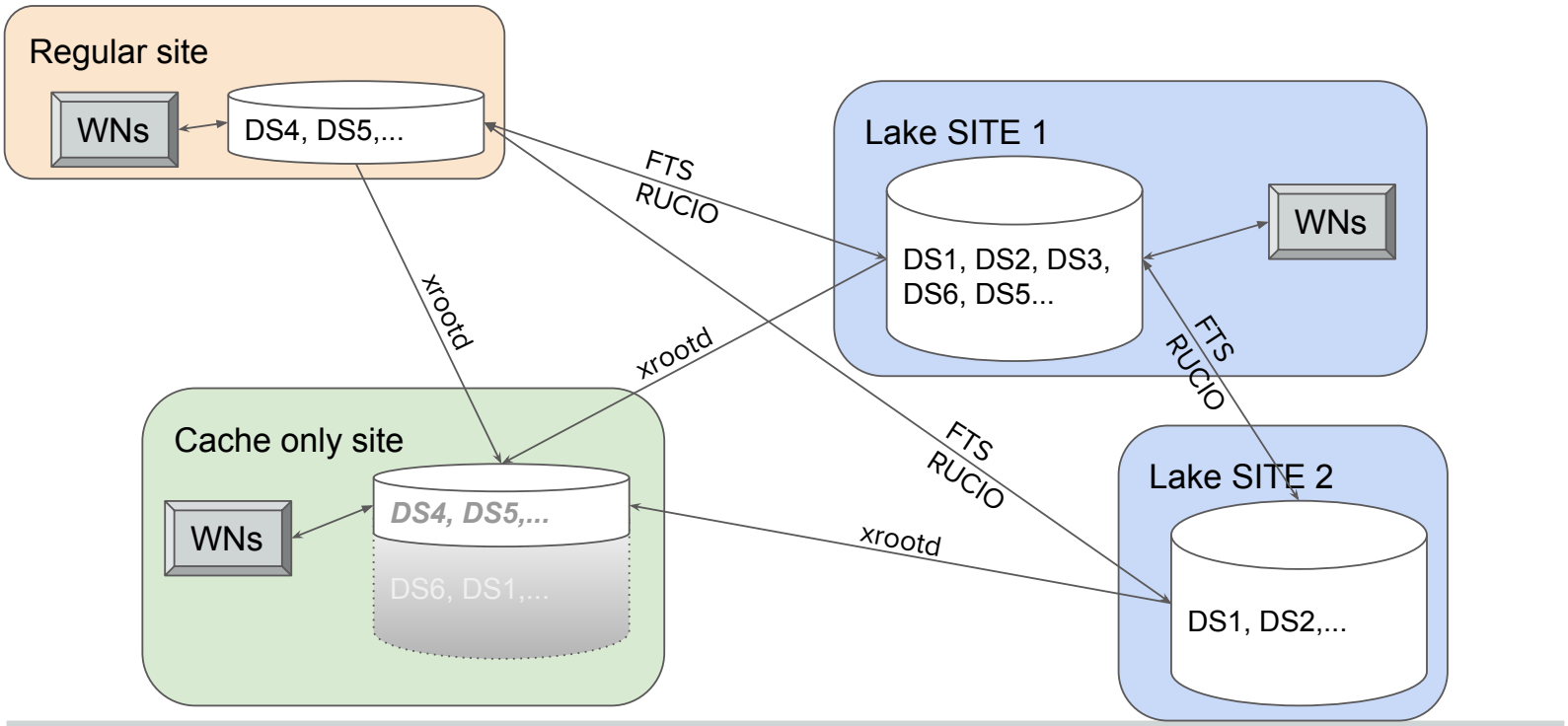
Other activities - open questions

- DOMA/ACCESS: Contributing to a document that is supposed to become a white paper with recommendations for HL-LHC
- New analysis formats play a role in the discussion (MiniAOD, NanoAOD)
  $\rightarrow$ unclear where caching will play a role
  $\rightarrow$ some believe the smallest formats will be stored on institute disks, some believe in "analysis facilities"
- Probably talking about different things:
  - Analysis facilities, very small formats: Caching in addition to storage for fast access on local computing resources (what Karlsruhe wants to do?)
  - Caching for "diskless" sites (context we studied so far) in grid

Virtual placement - Slide by Ilija Vukotic:

What is it actually ?

DSX - primary copy
DSX - virtual copy fully or partially cached data
DSX - virtual copy - not there at all until needed



3

→ "virtually" place datasets to cache-only sites
→ expected to ensure high hit rates
→ service set up by Ilija, test it in Munich?

Backup