ATLAS Data Carousel R&D

Xin Zhao

WLCG Archival Storage WG

July 5th, 2018

BROOKHAVEN NATIONAL LABORATORY

U.S. DEPARTMENT OF ENERGY

Outline

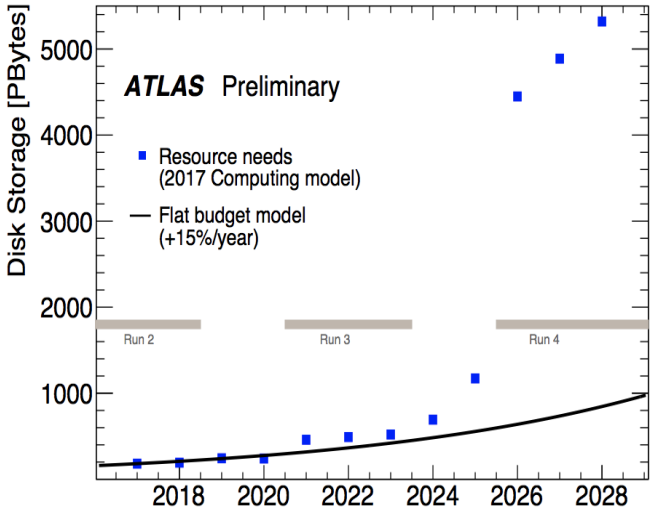
- Introduction
- The Plan
- Phase 0 tape performance test
  - Review of the BNL test
- Next step(s)

** This talk is based on discussion with ADC management and contributions from many ADC and sites experts*

Introduction (1/3) : HL-LHC

ATLAS perspective on the data storage challenge of HL-LHC:

- ‘Opportunistic storage’ basically doesn’t exist
- Format size reduction and data compression are both long-term goals, require significant efforts from the software and distributed computing teams
- Tape storage is 3~5 times cheaper than disk storage, increasing tape usage is a natural way to cut into the gap of storage shortage for HL-LHC

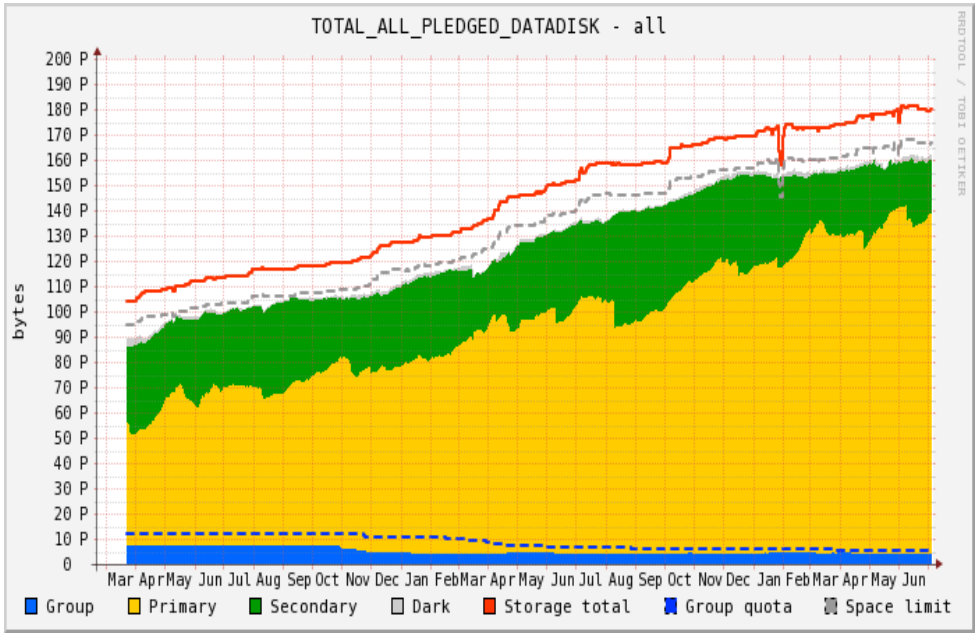


Exploit more tape usage

Introduction (2/3) : mid-term

ADC DDM operations: continuously fighting for space

AODs (Analysis Object Data) take 30% of disk space. Can we put all AODs on tape (even before Run-3) ???



Exploit more tape usage

Introduction (3/3) : Data Carousel R&D

- to study the feasibility to run various ATLAS workloads from tape
  - Start with derivation workload: its inputs are AODs

- *By 'data carousel' we mean an orchestration between workflow management (WFMS), data management (DDM/Rucio) and tape services whereby a bulk production campaign with its inputs resident on tape, is executed by staging and promptly processing a sliding window of X% (5%?, 10%?) of inputs onto buffer disk, such that only ~ X% of inputs are pinned on disk at any one time.*

The Plan (1/4) : Objectives

- Rucio
  - improve tape usage, e.g. bulk request to tape, with size tailored to site parameters
- FTS
  - optimize scheduling of transfers between tape and other storage endpoints, e.g. dedicated FTS instance for tape recall
- SE endpoints (dCache, Storm, etc) :
  - any bottlenecks and possible improvements on interfacing with respective tape backend ?
- Optimize data placement to tape
  - "do writing right" is the key ?
  - Use tape familities for files to be read back multiple times
  - Larger file sizes (10GB+ preferred)
- Evolving tape scheduler
  - Support high priority, low latency request
- PS2
  - study and optimize prompt processing of data as it appears off of tape --- process immediately when X% (5%? 10%?) of a dataset is staged ?
- WLCG Archival Storage WG
  - Work together, define realistic expectations and evaluate possible evolution

List of possible R&D topics (in no particular order), touches all areas of ADC

The Plan (2/4) : Objectives

- Rucio
  - improve tape usage, e.g. bulk request to tape, with size tailored to site parameters
- FTS
  - optimize scheduling of transfers between tape and other storage endpoints, e.g. dedicated FTS instance for tape recall
- SE endpoints (dCache, Storm, etc) :
  - any bottlenecks and possible improvements on interfacing with respective tape backend ?
- Optimize data placement to tape
  - "do writing right" is the key ?
  - Use tape familities for files to be read back multiple times
  - Larger file sizes (10GB+ preferred)
- Evolving tape scheduler
  - Support high priority, low latency request
- PS2
  - study and optimize prompt processing of data as it appears off of tape --- process immediately when X% (5%? 10%?) of a dataset is staged ?
- WLCG Archival Storage WG
  - Work together, define realistic expectations and evaluate possible evolution

"Tape Carousel" discussion points

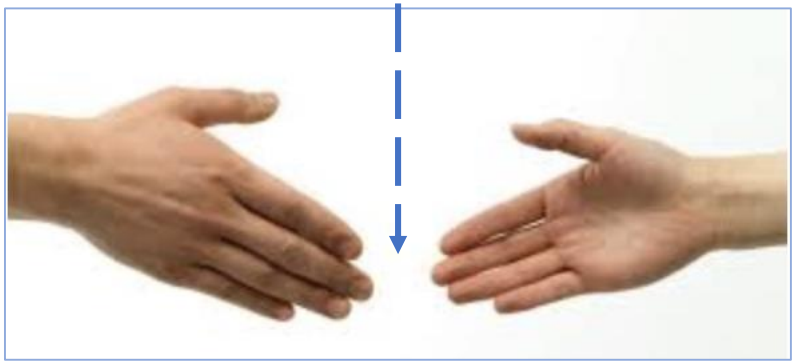
- Experiments should not get direct access to tapes
- *We* (the tape site managers) should try to improve the overall throughput of the infrastructure over time, but the experiments shall not make any assumptions on collocation, ordering nor time to completion when retrieving data from tape
- In fact "temporal collocation" of data does not hold in the long run
  - Can be configured and achieved in year X, but X+5 or X+7 years later:
    - The hot data becomes cold
    - Data is usually reshuffled across different tapes due to repack
    - The only collocation that might work is the one within the (larger) file
- More tape drives will be needed for sustained recall activity
  - Total cost of the tape infrastructure will increase
- Fast (but small-ish) buffer in front of tape (SSD based?) will be needed
  - SSD price decrease will continue, but unlikely to get significantly close to the HDDs
- Bulk prestaging is key for allowing internal optimisation of requests (maximising throughput, minimising tape mounts)
- *We* (the tape site managers) need to define what "tape carousel" means depending on what is feasible

WLCG Worldwide LHC Computing Grid

10

The Plan (3/4) : Expectation Management

- Experiments vs Tape services
  - not tug-of-war, but handshaking



  - ‘data carousel’ vs ‘tape carousel’

The Plan (4/4) : Three Phases

- First phase
  - Understand tape system performance at various T1 sites
    - T0 ?
  - Identify workloads (start with derivation), and evaluate performance based on current systems
    - Tape available at ~ 10 sites, while processing happens everywhere
    - Performance with tape vs disk
- Second phase
  - Address issues found in phase 1
  - Deeper integration between workload and data management systems (PanDA/PS2/Rucio)
- Third phase
  - Integrate with production system and run production, at scale, for selected workflows

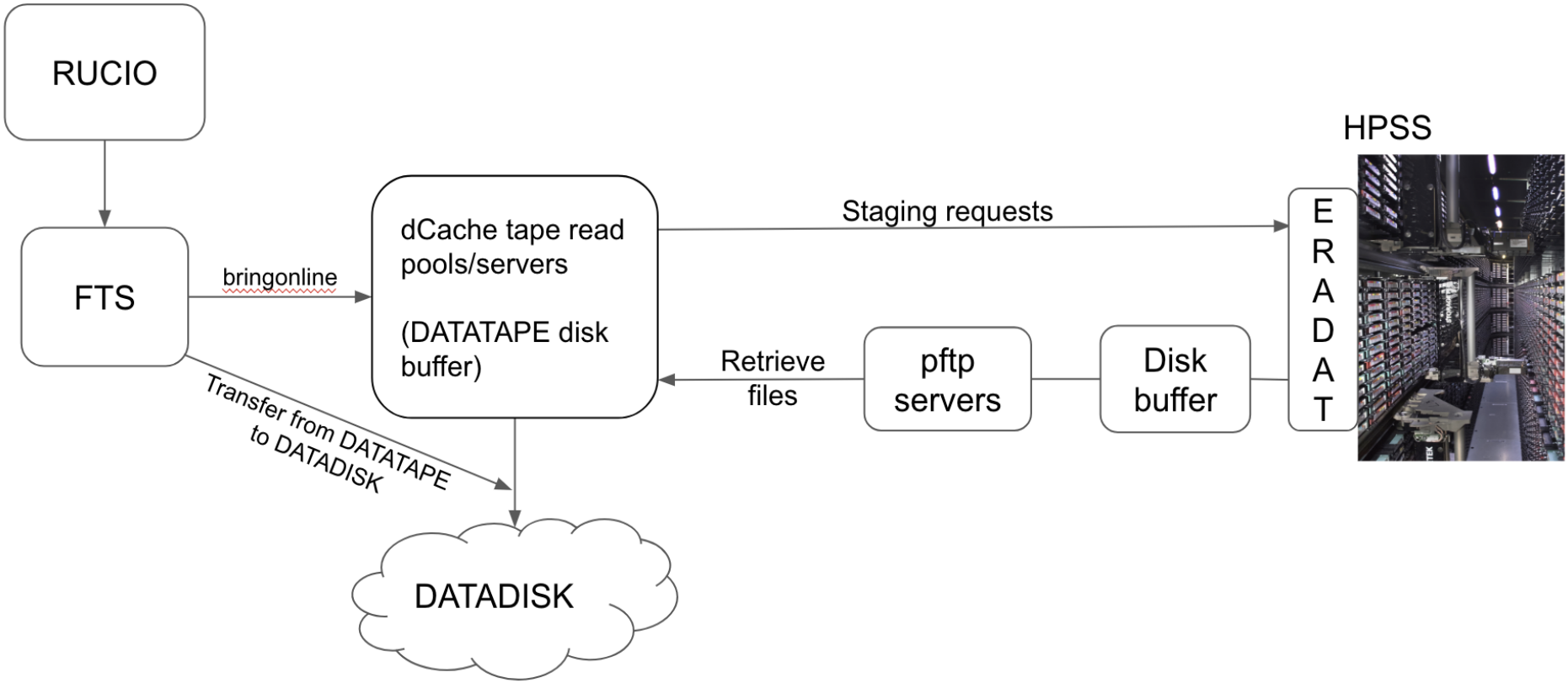
Phase 0 : Tape Performance Test

- Focus on tape system itself, measure its performance under current production condition
  - Establish a baseline measurement, as the starting point
  - Cover as many T1 sites as possible, for a better picture of the landscape
    - Each T1 tape system is different, different tape hardware, different batch system, also serving different user bases …
- Started with BNL
  - The following several slides are a review of it …

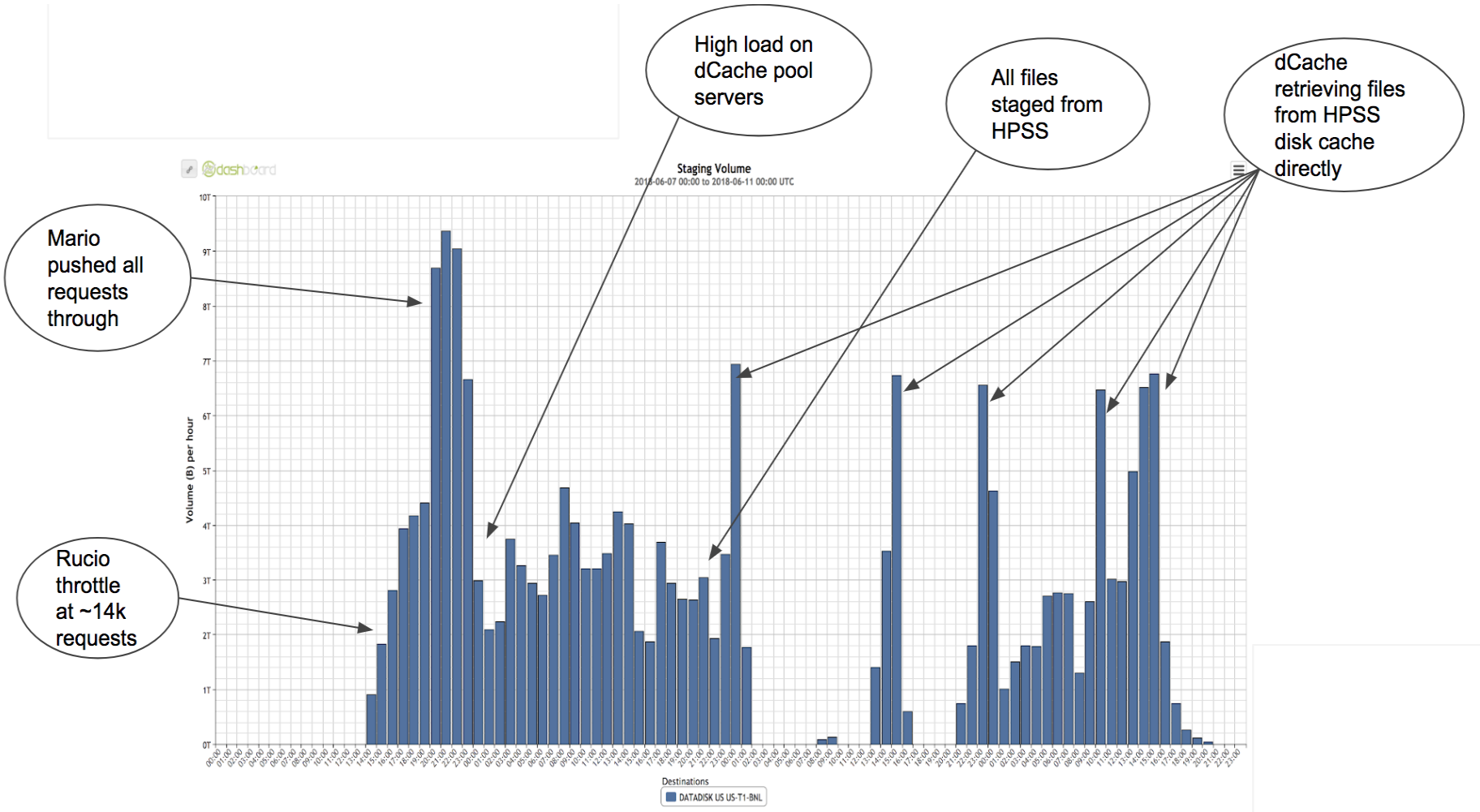
BNL test : test setup

- Staging test
  - BNL DATATAPE → BNL DATADISK
- Time window (~4 days)
  - 14pm (UTC), 06/07/2018 → 20pm (UTC), 06/11/2018
- Data sample
  - 33 AOD datasets: data16_13Tev
  - 110k files, 200TB, average file size ~ 2GB
    - Distributed among 547 LTO-6 tapes, ~200 files per tape, which is 16% of all data of one LTO-6 tape
    - Total size turned out to be too big for our current dCache configuration
- 31 tape drives used in HPSS
  - Normally use 22 (LTO-6/LTO-7) drives
  - Added more drives to push for the limit

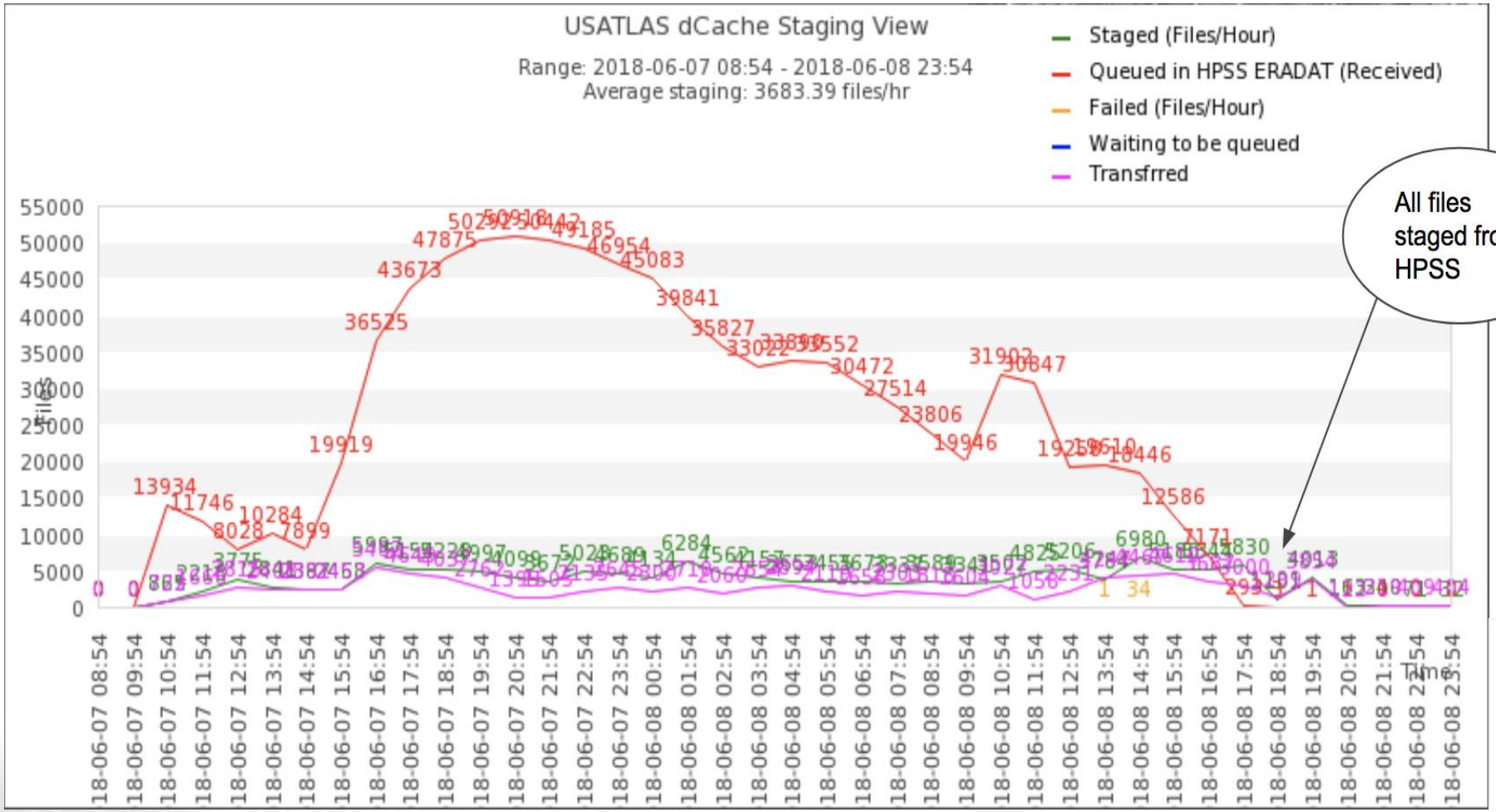
BNL test : system layout



BNL test : result on DDM

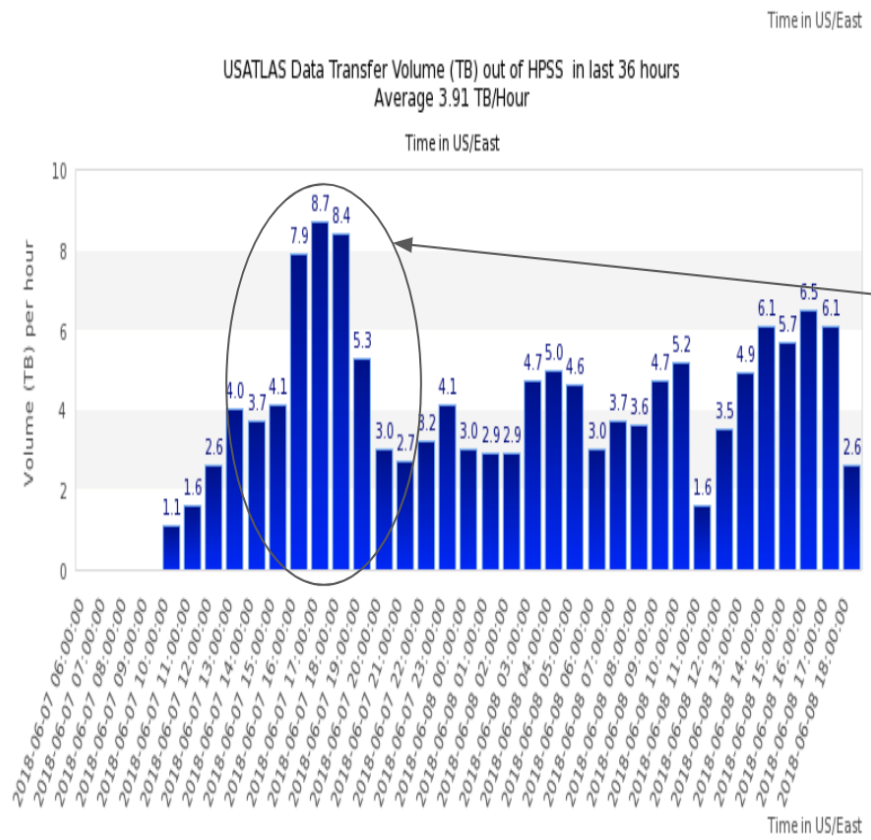


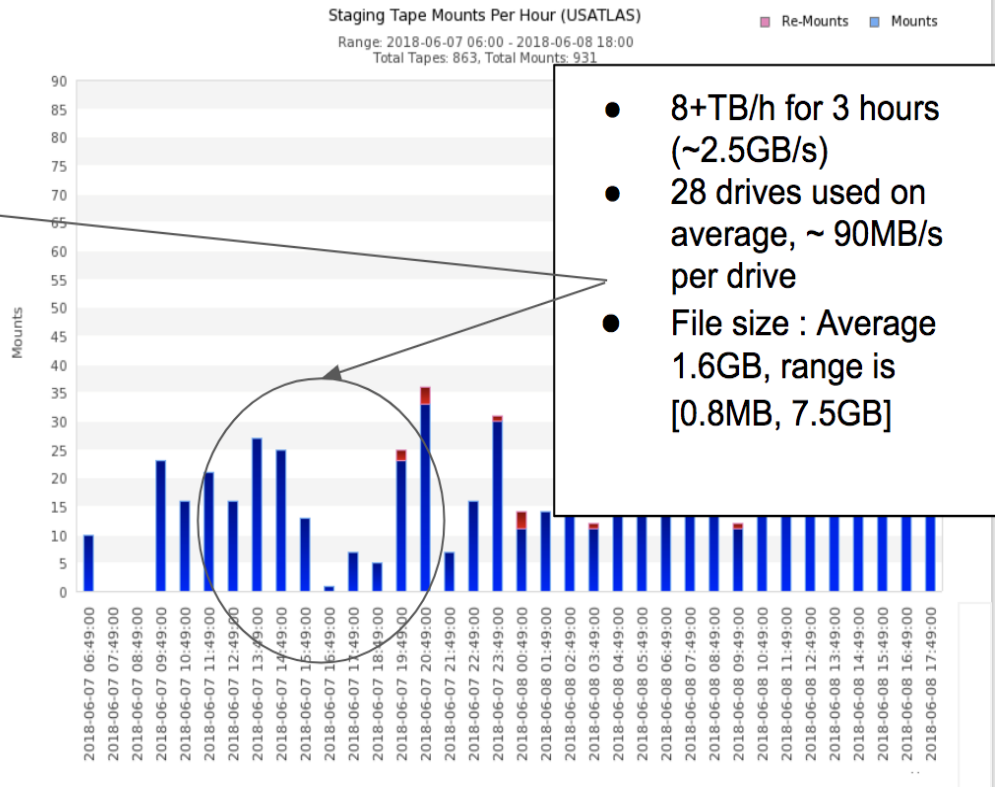
BNL test : result on HPSS



BNL test review : HPSS (1/2)

- Reached (current) maximum tape performance





- 8+TB/h for 3 hours (~2.5GB/s)
- 28 drives used on average, ~ 90MB/s per drive
- File size : Average 1.6GB, range is [0.8MB, 7.5GB]

BNL test review : HPSS (2/2)

- Current limitation is on disk buffer of HPSS
  - Disk I/O rate is 2.5GB/s one way at most
    - Purchasing new disk which can do 10GB/s one way
- Other factors (secondary)
  - All data are on LTO-6 tapes, LTO-7 can give better (double) performance, which can be considered in future tests
  - Cap on the number of concurrent pftp connections for data retrieval from HPSS disk buffer
    - Not dominant factor in this test, but will increase it from 600 to 3000, for future tests

BNL test review : dCache

- Number of dCache tape read pool servers
  - Not enough read pool servers for active requests at O(100k), causing high load and slow retrieval from HPSS disk buffer
    - Had to decrease polling rate (every 1m → every 10m) to reduce load
- Space of dCache disk buffer on tape read pools
  - Usable space is ~150TB, too small for the data sample, resulting in queuing requests on pinmanager, because all disk space already reserved by earlier requests
- Plan in place already
  - Increase size of dCache disk buffer to O(PB)
  - Increase number of tape read pool servers
- ***To dCache developers***
  - To help reduce the load on dCache servers, consider to use a callback mechanism for tape recall, instead of polling from dCache side ?
  - Improvement on space reservation mechanism on the dCache read pools, to help reduce demand on size of disk buffer ?

BNL test review : Rucio

- Rucio throttles requests at the beginning, needed manual intervention to push them through
  - Easy fix: not using the right CLI option to submit the requests.
- Rucio submits requests (files) from each dataset proportionally, meaning the progress of file stage is in parallel among all datasets, no matter how big the dataset is.
  - Result: all datasets suffer the same long tail

Next Steps

- Tape test on other T1 sites ?
  - INFN, PIC, FZK, …
- Repeat the test at BNL
  - Tune dCache and pftp parameters
  - More importantly, wait after new hardware to arrive (ETA: end of summer)
- Move to phase 1 of ‘data carousel’ R&D
  - Involving derivation tasks, tape recalls will be driven by jobs
  - Compare task completion time with data from tape vs from disk