BNL’s experience on MaS and Data carousel

Hironori Ito

On behalf of Yinzi Wu, Zhenping Liu, Xin Zhao, Eric Lançon

Brookhaven National Laboratory

BROOKHAVEN NATIONAL LABORATORY

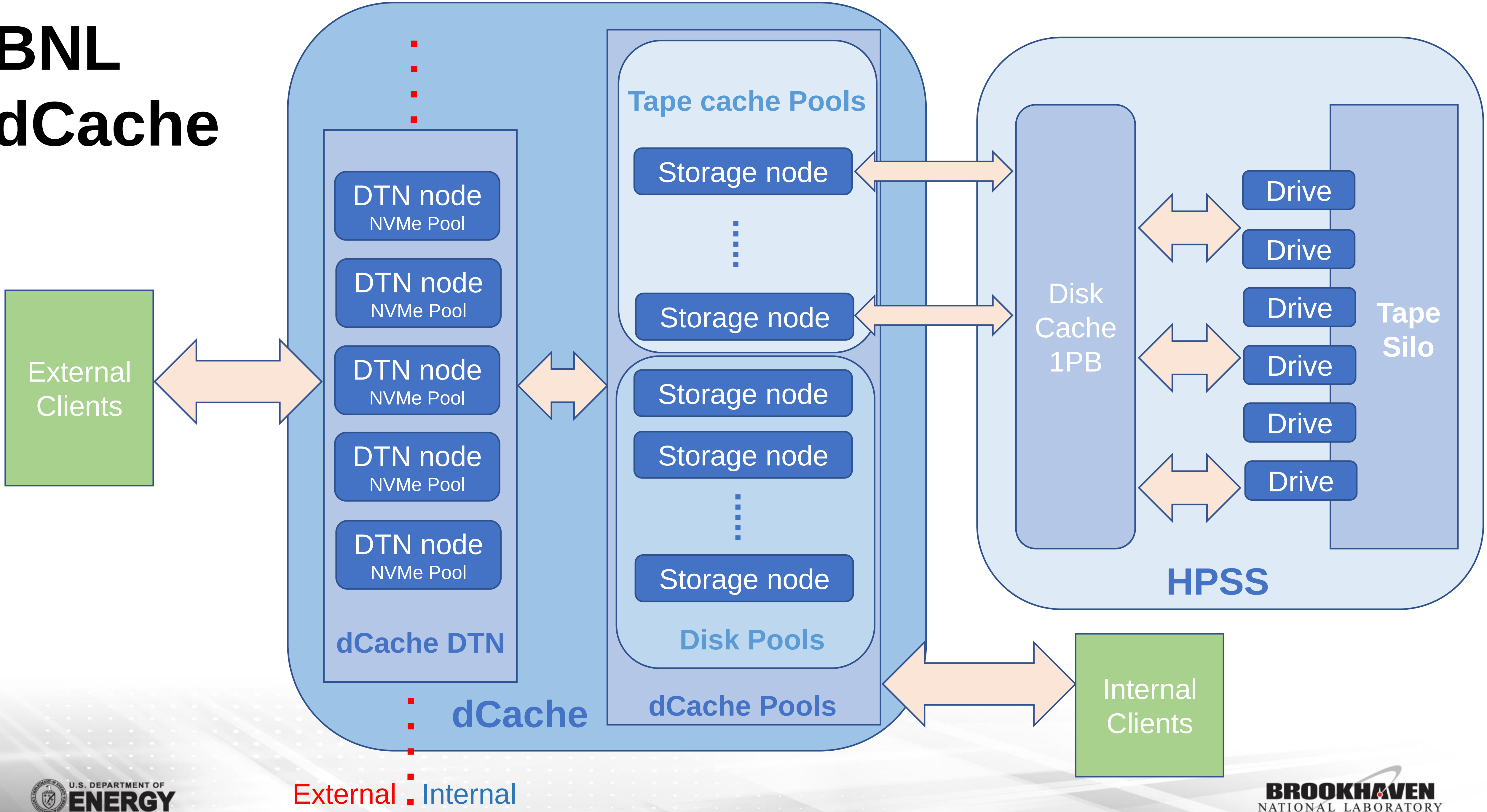
U.S. DEPARTMENT OF ENERGY

BROOKHAVEN SCIENCE ASSOCIATES

BNL Storage System; dCache and HPSS

- dCache
  - 18 DTN nodes 2x10 Gbps
  - 53 storage nodes
    - 2 x 10 Gbps or 2 x 25Gbps in newer hosts.
    - The size from 0.5PB to 1.2PB
  - Large disk cache for tape read requests
    - 5PB compare with the typical size of the disk cache ~100s TB (BNL had 200TB before substantial increase)
- HPSS
  - 30 LTO-7 drives
  - 1PB disk cache

BNL dCache



External Clients
DTN node
NVMe Pool
DTN node
NVMe Pool
DTN node
NVMe Pool
DTN node
NVMe Pool
DTN node
NVMe Pool
dCache DTN
Tape cache Pools
Storage node
Storage node
Storage node
Storage node
Storage node
Disk Pools
dCache Pools
dCache
Disk Cache 1PB
Drive
Drive
Drive
Drive
Drive
Drive
Tape Silo
HPSS
Internal Clients
External
Internal

How data are written to the tape system in HPSS

- Files written to HPSS **disk cache** are written to the tapes in the order that were written to HPSS disk cache FIFO.
- Files assigned to all **write** drives. The files are sprayed to all write drives.
- However, all **write** drives have the same file family when files are written. Files belonging to different file family will wait until the tapes belonging to their file family are mounted.
- Writing to tape happens only when the usage of disk cache is more than the certain level, water-mark, or preset time once a day.
- File family (aka tape set) can be used to isolate the group of the files.
  - It must be pre-created

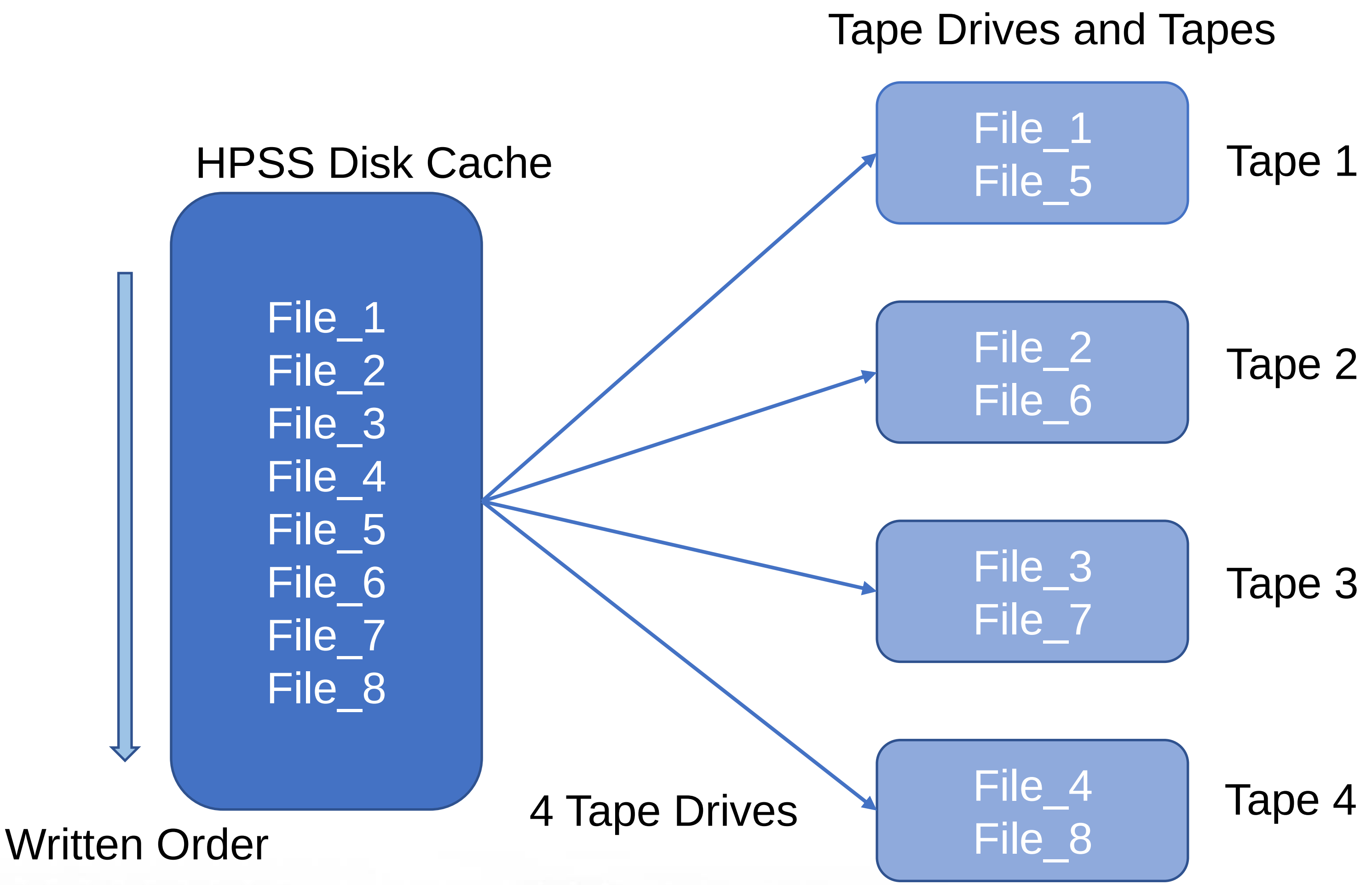
How ATLAS stores Data in Tape

- BASEPATH/scope/type/metadata/datasetname/files
- File family (aka tape set) is created in BASEPATH/scope level
  - Scope level is chosen because
    - Scope/type (or anything below that directory) level might contain too small amount of the data, leaving too many empty tapes.
    - Too many of them might have operational issue.
    - It requires pre-population of the file family.

Simple case with one file family

Assumption

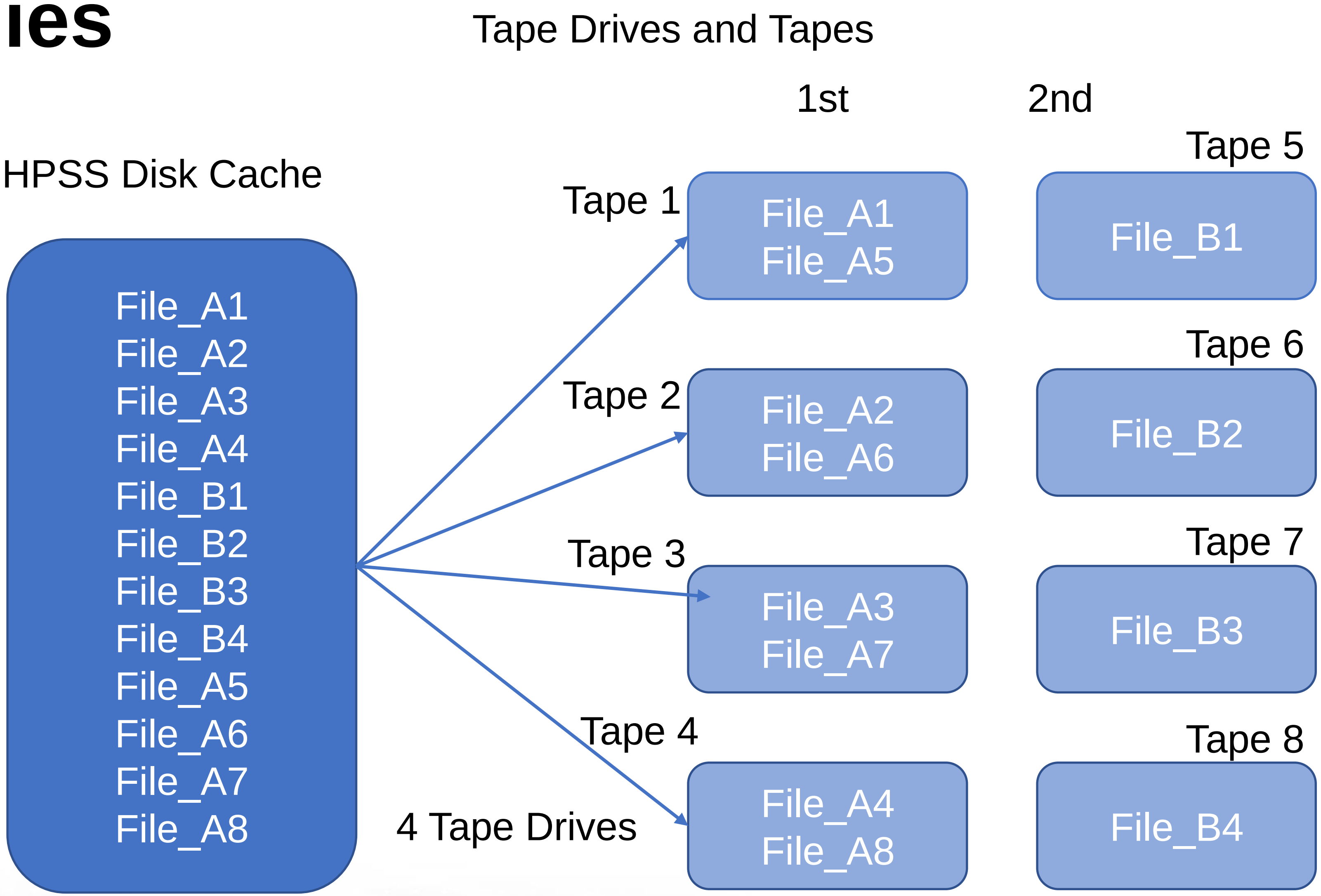
- 4 tape drives are assigned for write.
- 8 files are written to HPSS disk cache in the numeric order shown.
- All 8 files belong to the same file family.
- Files are written to the cache area within the short time. And, the sizes of files are large enough to require the use of all four write drives.



Multiple file families

Assumption

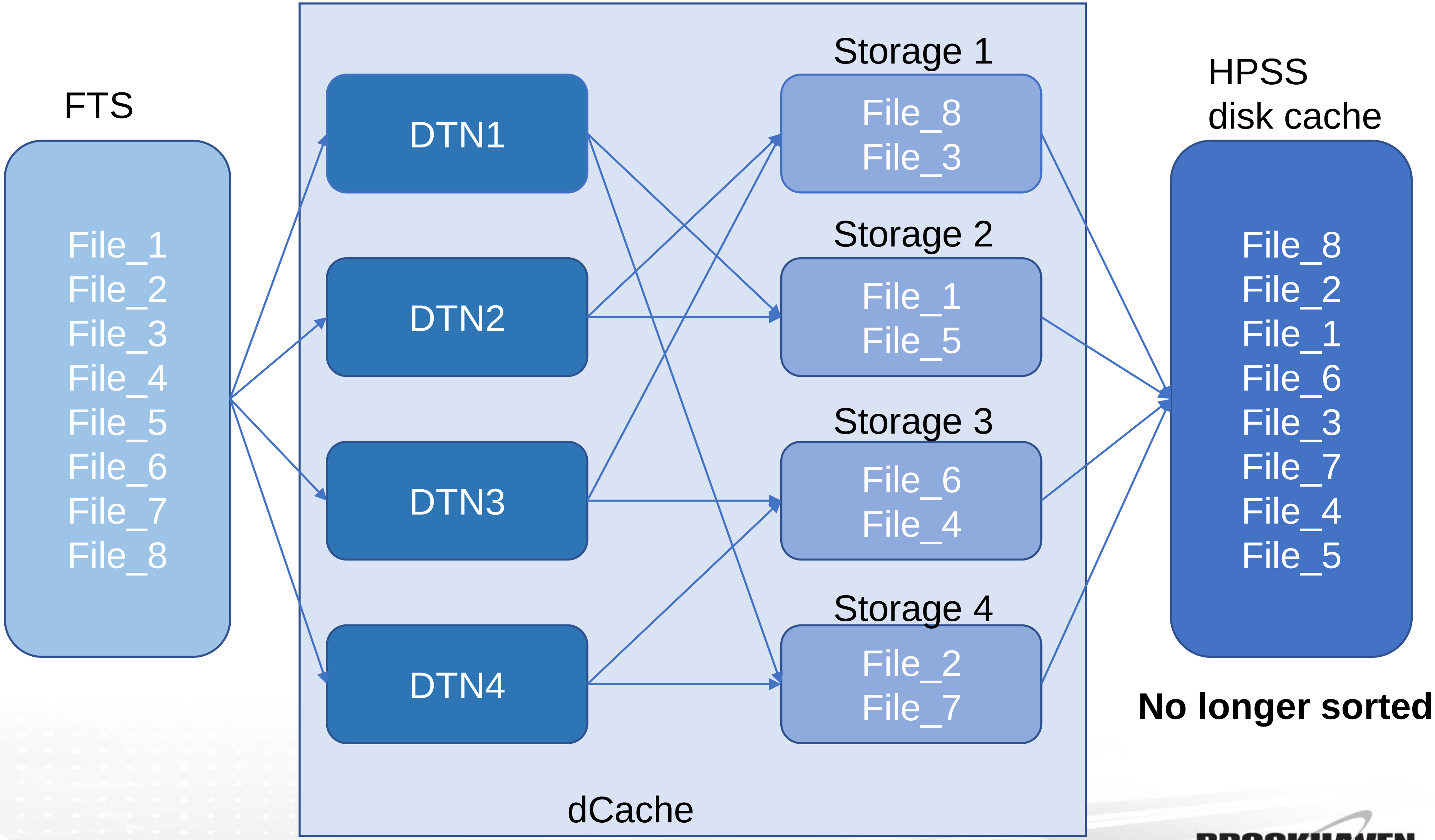
- 4 tape drives are assigned for write.
- 12 files are written to HPSS disk cache in the numeric order shown.
- File_A 8 files belong to the same file family while File_B 4 files belong to the different file family
- Files are written to the cache area within the short time. And, the sizes of files are large enough to require the use of all four write drives.



How files are written to HPSS disk through dCache

Assumption

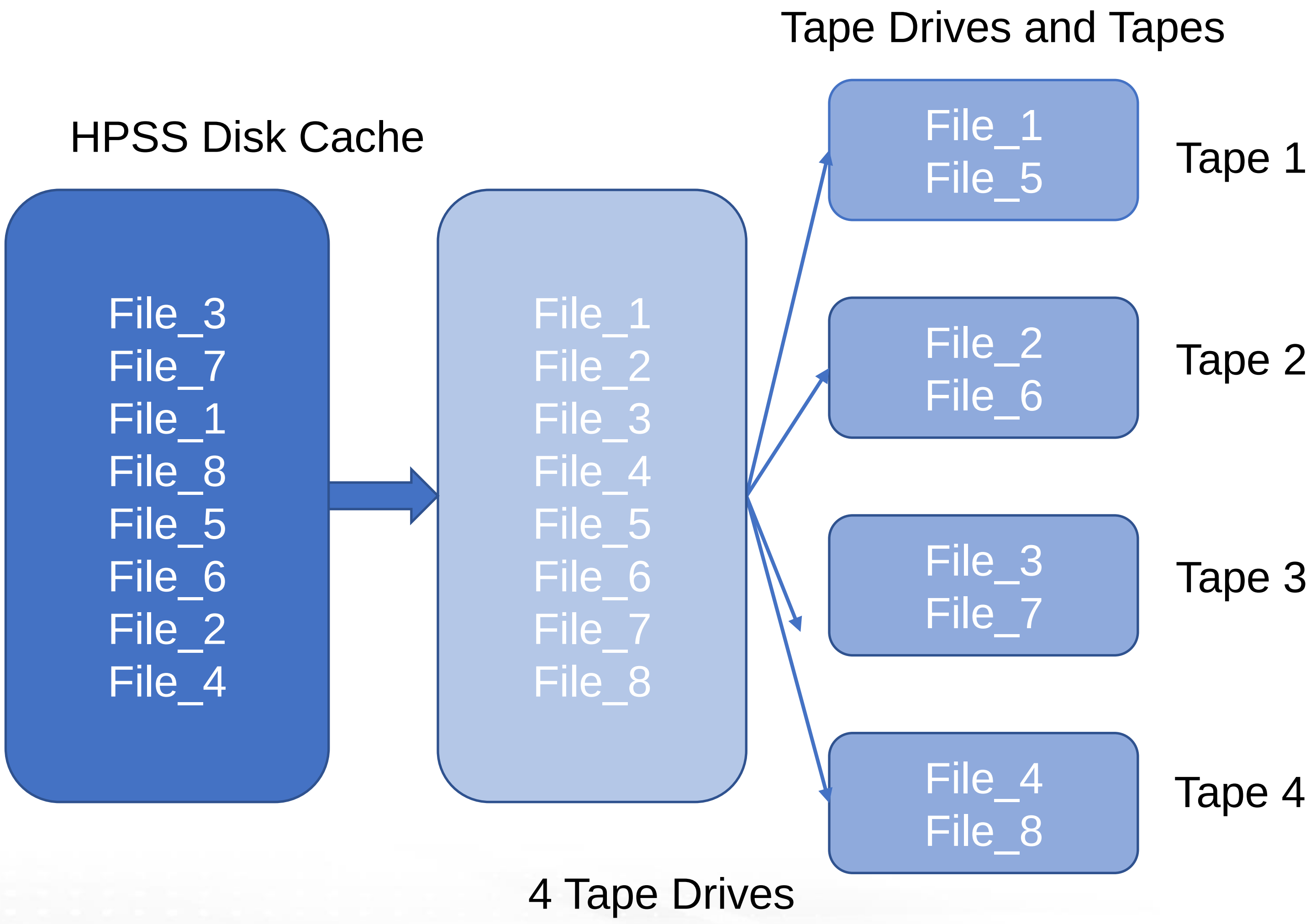
- FTS transfers multiple files concurrently.
- DTNs are selected based on load/performance.
- Storage is selected based on the load/performance/spaces.
- The storage of the source of the files are not necessary from the same or similar performance.
- The variations in the transfer time could be very large.



Directory based sorted write of new HPSS

Assumption

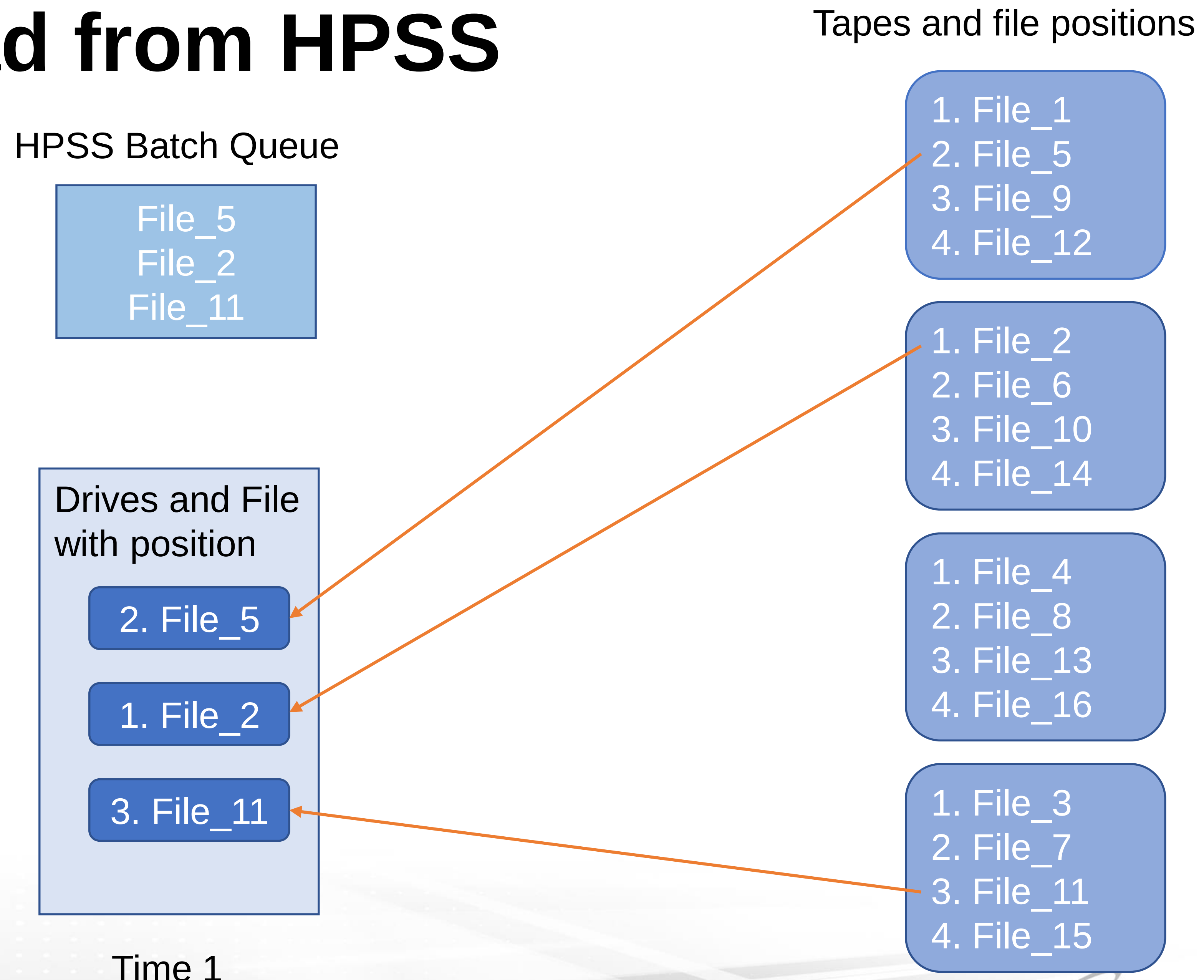
- 4 tape drives are assigned for write.
- 8 files in the **one directory** are written to HPSS disk cache in the **random** order.
- All 8 files belong to the same file family.
- Files are written to the cache area within the short time. And, the sizes of files are large enough to require the use of all four write drives.



How files are read from HPSS

Assumption

- 3 drives are available for read.
- 16 files will be read at total.
- Files written to 4 different tapes.
- At first, File_5, File_2 and File_11 requests arrived to HPSS Queue.



Time 1

How files are read from HPSS

Assumption

- 3 drives are available for read.
- 16 files will be read at total.
- Files written to 4 different tapes.
- At first, File_5, File_2 and File_11 requests arrived to HPSS Queue. (There is no guarantee that the 1st file to read on the tape has the lowest file mark.)

HPSS Batch Queue

Time 1

File_5
File_2
File_11

Drives and File with position

- 2. File_5
- 1. File_2
- 3. File_11

Time 2

File_5
File_2
File_11
File_1
File_6
File_12
File_15
File_3
File_4
File_8
File_10
File_9
File_7
File_13
File_14

After Sort

Rewind
2. File_5
1. File_1
3. File_9
4. File_12

1. File_2
2. File_6
3. File_10
4. File_14

Rewind
3. File_11
1. File_3
2. File_7
4. File_15

Waiting for drive
1. File_4
2. File_8
3. File_13
4. File_16

Tapes and file positions

1. File_1
2. File_5
3. File_9
4. File_12

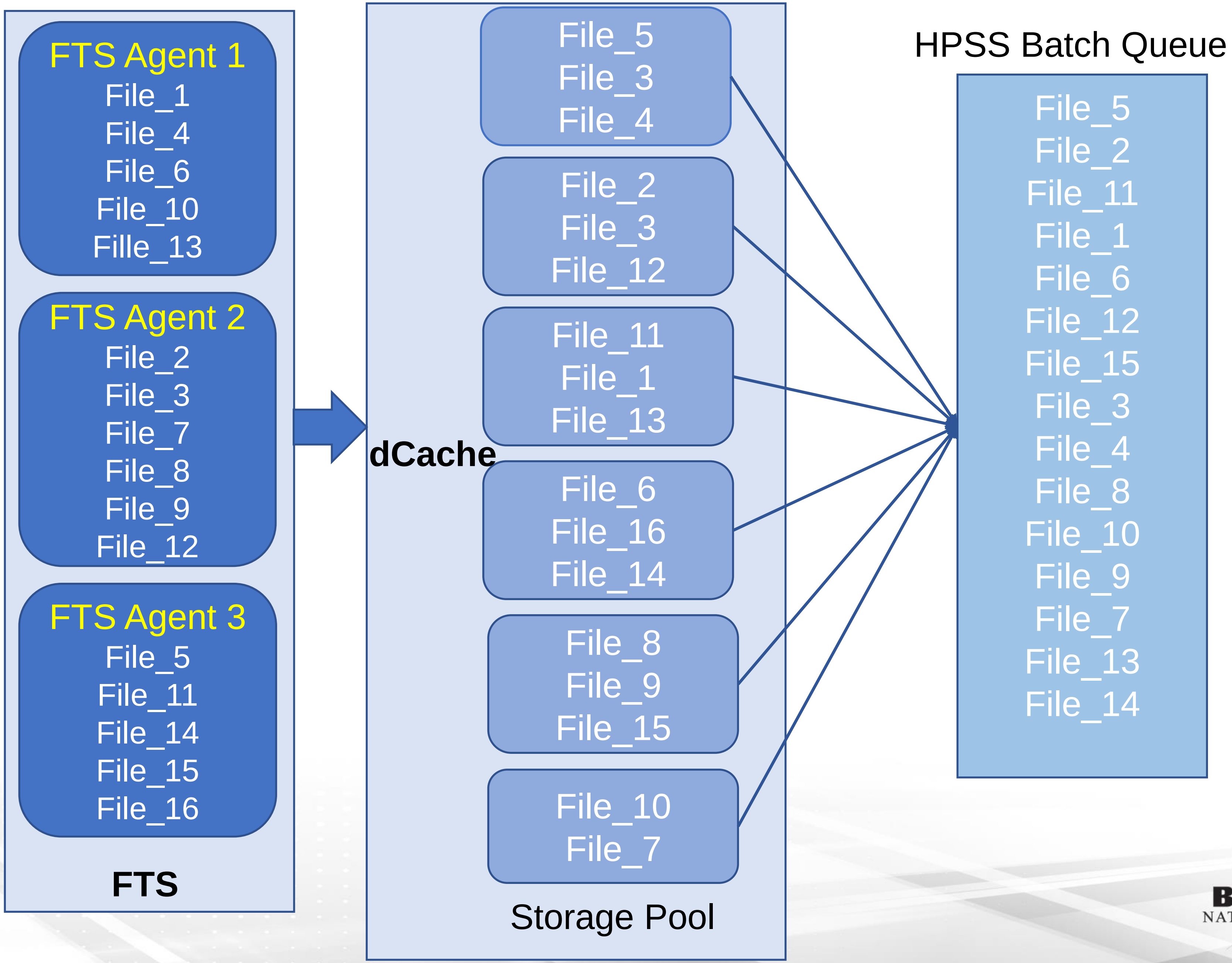
1. File_2
2. File_6
3. File_10
4. File_14

1. File_4
2. File_8
3. File_13
4. File_16

1. File_3
2. File_7
3. File_11
4. File_15

How requests arrive to HPSS Batch

- RUCIO will request to FTS on file-by-file basis.



Real example

- Example is taken from ATLAS RAW DATA.
- The file size is small (<1GB) in this example.

One dataset
4818 files
8 tapes
- 1055(/10019)
- 252(/5191)
- 114(/4303)
- 290(/5856)
- 247(/6070)
- 1251(/8746)
- 787(/10485)
- 822(/9941)

3295: File_0995
3301: File_1191
3302: File_0101
3303: File_0450
3306: File_4558
3307: File_1700
3323: File_0299
…

Small gap of 5 files

Small gap of 2 files

Medium gap of 15 files

3295: 1/6
3301: 1
3302: 1
3303: 1/3
3306: 1
3307: 1/X
3323: 1…
…

Small gap: Tape moves at the same speed for forwarding without disengaging from the head.
Every small skip of N files reduces the effective throughput by factor of N+1.
1-file small gap (1/2)
2-files small gap (1/3)

Medium (or large) gap: Tape moves at fast speed for forwarding after the head is disengaged.

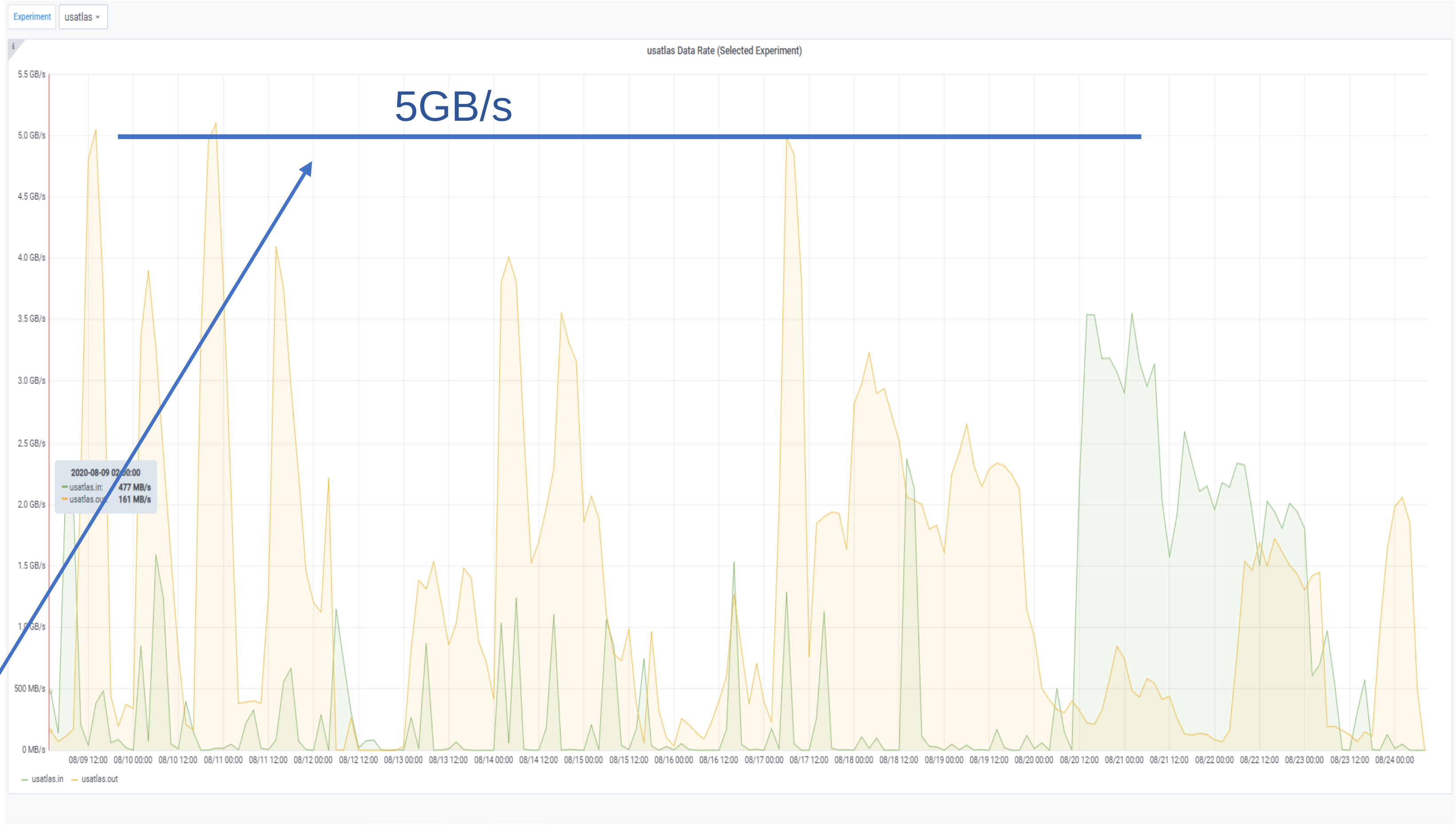
Rewind: same as Big gap

Total effective throughput = (1/6+1+1+1/3 +1+1/x + 1)/7 -> **0.57** (assume x is large)

File gap in a tape reduces the throughput quickly.
New version of HPSS with <u>sorted write</u> should help the overall throughput on read by eliminating all small and medium size file gaps**.**

Real Data Rate seen in BNL HPSS

- The real data rate changes greatly by the number of assigned drives, number of file gaps, the size of file gaps, how many tapes, etc…



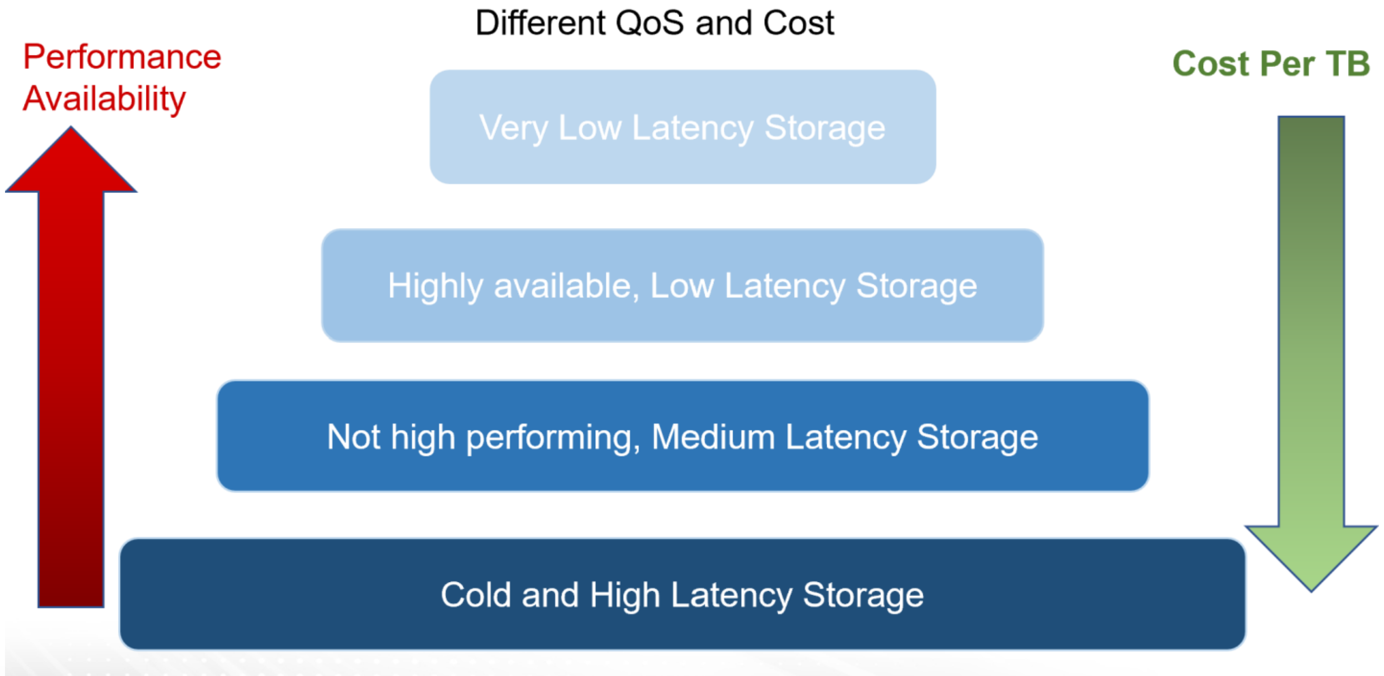
Despite all the possible issues, the rate at can exceed 5GB/s.

How to improve the transfer rate on read.

- Larger file size always help. Anything larger than a few GB will be close to the maximum rate on that file.
- Reduce the number of gaps.
  - Small and medium sized file-position gaps will be eliminated by the directory-based, sorted-write feature of new version of HPSS. It will be deployed later in 2021 at BNL. NOTE: The feature is already available in HPSS.
  - The larger file position gap will be only eliminated if all files in a directory are written in the short period.   It needs to be short enough that when the last files in a directory is written to HPSS disk cache, the other files in the directory are still in the queue. Bulk writing is important.
- Read-requests come in bulk to HPSS cache.
  - Make sure to read all files in that directory.
    - Maybe, we can make it default. If the number of requests in a directory is more than N files (or M %), we should just read/stage them all.

Multilayer Automated Storage, MaS

- Investigation of storage cost reduction by introducing an intermediate storage class between disk and tape
  - Trade high performance disk storage for tape & low cost disk storage
  - High-cost disk storage reserved for frequently used and high value data
  - Other data are either on low-cost disk & tape or on tape only
  - Active data migration between various storage classes

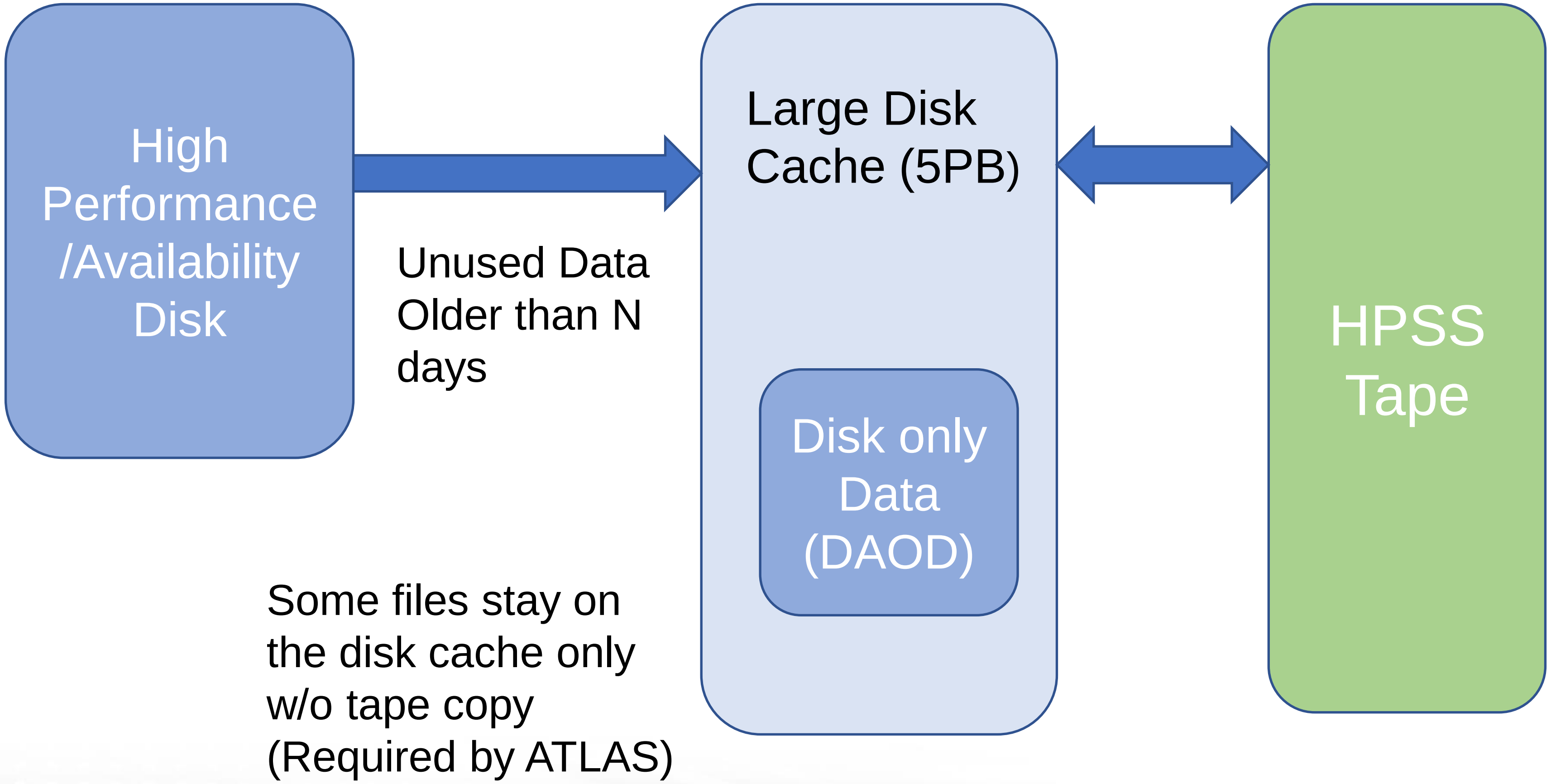


Efficient use of storage.

- Large fractions of disks data are not accessed often.
  - For an example, ~30% of volume of the data on the high performance disks are not read more than 100.
- Storing the unused data on the precious, expensive, limited volume of disks, is not cost-efficient way of using disks.
  - Different types of storage are available for cold(er) data.
- Some data are used heavily.
  - Different types of higher performing storage are also available.

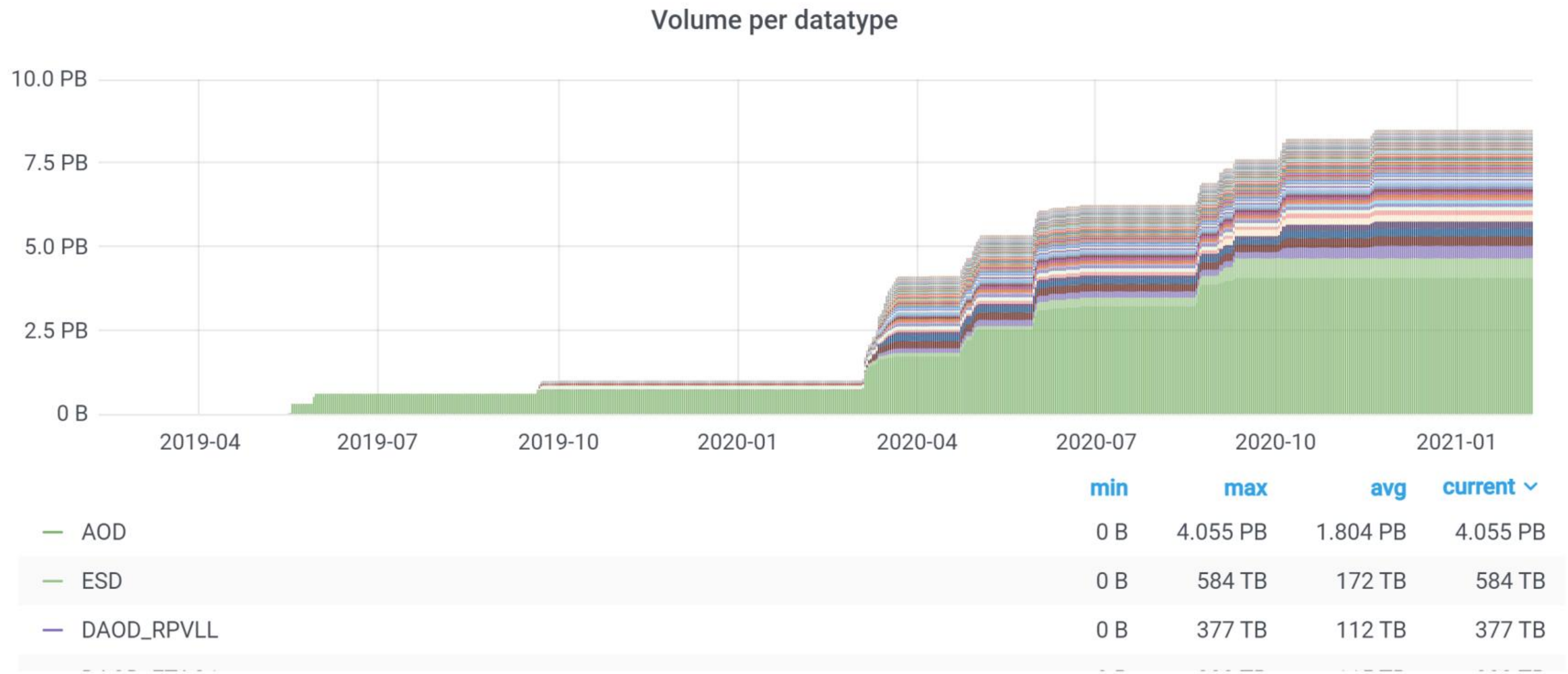
Data Movements for MaS

- BNL has setup very large disk cache space (5PB)
- Unused data on high performance / availability storage are transferred to tape-backed area.
- Data on MaS is used for the production.



Data growth in MaS storage endpoint

- 8PB of the data have been moved, creating more space for necessary data.



Conclusion

- File location gaps in tape slows down the read throughput.
- New version of HPSS will eliminate small and medium size gaps in tape.
    - Will be deployed later in 2021
- The large file gap can be only eliminated if all files within one dataset are written to the tape cache within the time windows.
- MaS prototype will continue to take the data to evaluate the use of tape-backed layered storage in the production environment.