

Analysis of files in BNL Tape system and their expected throughput

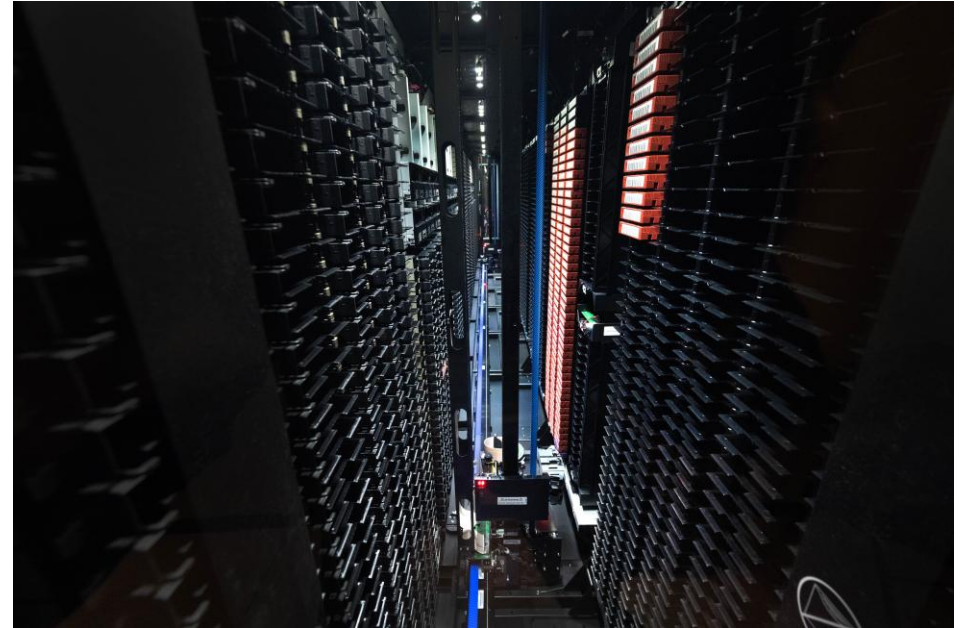
Hironori Ito

ATLAS TIM JAN 2024, Stony Brook University

Purpose of the presentation

IBM TS4500

- Analyze the structures of already written files in BNL HPSS tape system to see.
- Is the sorted-write feature enabled in Oct 2020 has any impact how files are written?
- Can the expected throughput be obtained from the knowledge of the structure of files in a tape?
- Can we conclude if anything else is needed?



Are we smart enough in the use of HPSS tape system?

Tape system

Data set (or songs you want in your playlist) A: A0, A1,

Pos	file	Gap	
30	A.0	-1	FF
31	B.3		Play
32	A.2	2	Play
33	A.3	1	Play
34	C.6		Play
35	D.2		Play
36	A.4	3	Play
37	A.5	1	Play
...	
137	A.6	100	FF
138	A.7	1	Play



Not FF

Remember creating a mixed tape in 70s~80s?
Remember how time consuming it was to pick **only** your favorite songs.



FF for big forward/gaps
Just let the music play for small gaps


In HPSS, this happens automatically.
FF kicks in for gaps larger than some numbers (10s? See later figure)

Simple model of the impact of gaps to the throughput (TH)

Data set A: A0, A1,

Good

Pos	file	Gap
30	A.0	-1
31	A.1	1
32	A.2	1
33	A.3	1
34	A.4	1
35	A.5	1
36	A.6	1
37	A.7	1
38	A.8	1
39	A.9	1
40	A.10	1




Average gap for A = 1

$$\text{ExpTH} = \text{TheoTH} / 1$$

Bad

Pos	file	Gap
30	A.0	-1
31	B.0	-1
32	A.1	2
33	B.1	2
34	A.2	2
35	B.2	2
36	A.3	2
37	B.3	2
38	A.4	2
39	B.4	2
40	A.5	2




Average gap for A = 2

$$\text{ExpTH} = \text{TheoTH} / 2$$

$$\text{Expected TH} = \frac{\text{Theoretical TH}}{\text{Average Gaps}}$$

Worse

Pos	file	Gap
30	A.0	-1
31	B.0	-1
32	C.0	-1
33	A.1	3
34	B.1	3
35	C.1	3
36	A.2	3
37	B.2	3
38	C.2	3
39	A.3	3
40	B.3	3



Average gap for A = 3

$$\text{ExpTH} = \text{TheoTH} / 3$$

Assuming the size of the files to be the same

If the size of files and the size of files in gaps are different,

$$ExpTH = \frac{\text{File Size in bytes}}{\text{Gap Size in bytes} + \text{File Size in bytee}} \times \text{TheoTH}$$

If file size is the same

$$ExpTH = \frac{\text{File Size}}{\text{File size} \times \text{Number of Gap}} \times \text{TheoTH}$$

$$ExpTH = \frac{1}{\text{Number of Gap}} \times \text{TheoTH}$$

Read Data from the LTO7 tape with gaps

A72084

Pos	Posdiff	TH (MB/s)	Size in bytes
18	1	315.801767	3.79E+09
19	1	285.4332488	3.711E+09
20	1	308.3705188	3.7E+09
21	1	314.9870533	3.78E+09
22	1	313.9359378	3.767E+09
23	1	231.5221325	3.704E+09
24	1	244.0708379	3.661E+09
25	1	312.8800798	3.755E+09
26	1	311.2494752	3.735E+09
27	1	306.1780514	3.674E+09

A73240

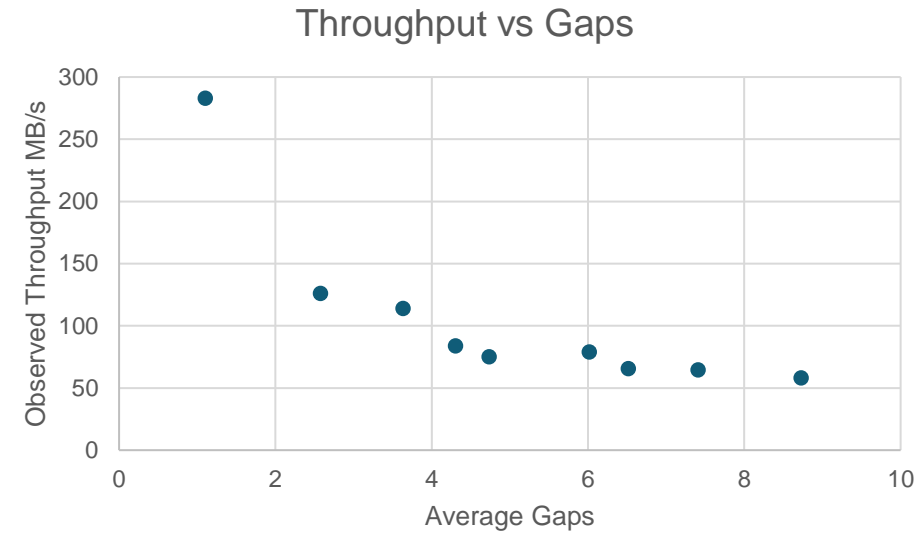
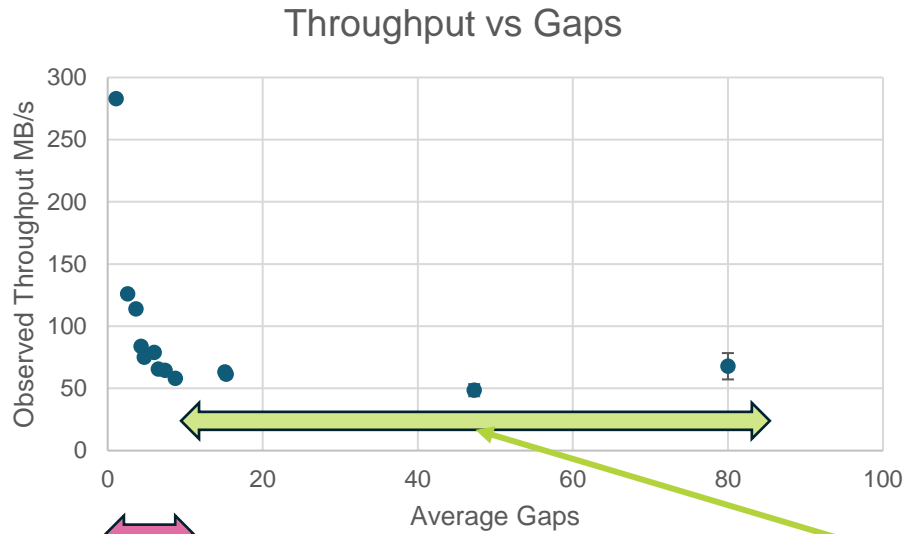
Pos	Posdiff	TH (MB/s)	Size in bytes
26	2	161.6649	5.01E+09
28	2	161.4812	5.01E+09
30	2	157.1033	5.03E+09
32	2	152.3197	5.03E+09
34	2	128.9106	5.03E+09
36	2	157.153	5.03E+09
38	2	163.2414	5.06E+09
40	2	152.8997	5.05E+09
42	2	157.1037	5.03E+09
44	2	142.9263	5E+09

A73240

Pos	Posdiff	TH (MB/s)	Size in bytes
192	3	79.9442	2.24E+09
195	3	126.8062	3.93E+09
198	3	119.7078	3.95E+09
201	3	93.47232	2.24E+09
204	3	44.94299	2.25E+09
207	3	77.38661	2.24E+09
213	6	42.0713	2.23E+09
216	3	148.1892	5.63E+09
222	6	33.44498	2.24E+09
225	3	97.25305	2.24E+09

- Various ATLAS LTO 7 tapes were read from the beginning to the end with various gap size to study the effect of the gaps
- File position and throughput are obtained from HPSS.
- The throughput shown is the value to HPSS disk cache from the tape drives.
- Gap sizes of 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 20, 40, 80 were tested.
- Sometime, the gap size is not what it is intended because
 - The requested files were already in the disk cache.
 - Files in the gaps are requested by something else (production system)

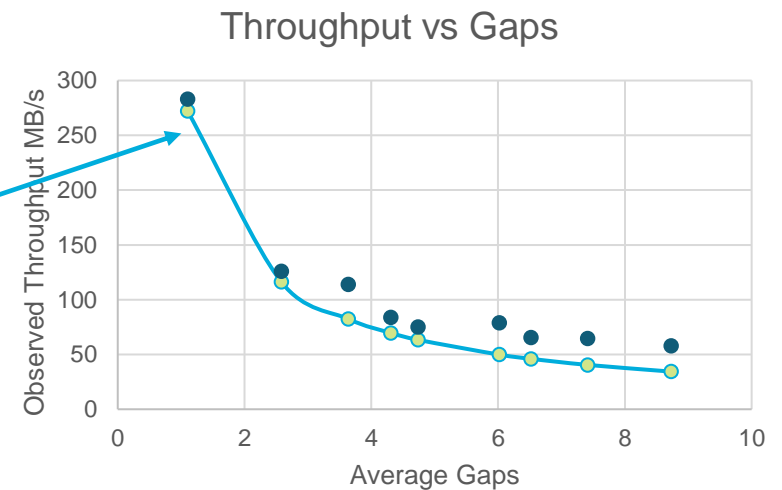
Results of actual throughput vs gaps



It is flat for gaps larger than 10

The earlier simple model seems to follow the observed behavior up to the gap size of about 5.

Simple model
 $TH = \text{Max TH} / \text{Gaps}$



ATLAS tape files in dCache

- ATLAS files that are written to tape space are grouped by the directory with their dataset name.
- data22_13p6TeV.00430183.physics_Main.merge.AOD.r14190_p5449_tid31407878_00
 - /pnfs/usatlas.bnl.gov/BNLT1D0/data22_13p6TeV/AOD/r14190_p5449/data22_13p6TeV.00430183.physics_Main.merge.AOD.r14190_p5449_tid31407878_00/

```
[root@spar0108 ~]# ls /pnfs/usatlas.bnl.gov/BNLT1D0/data22_13p6TeV/AOD/r14190_p5449/data22_13p6TeV.00430183.physics_Main.merge.AOD.r14190_p5449_tid31407878_00/
AOD.31407878._000002.pool.root.1      AOD.31407878._000329.pool.root.1      AOD.31407878._000694.pool.root.1      AOD.31407878._000936.pool.root.1
AOD.31407878._000003.pool.root.1      AOD.31407878._000330.pool.root.1      AOD.31407878._000695.pool.root.1      AOD.31407878._000937.pool.root.1
AOD.31407878._000012.pool.root.1      AOD.31407878._000331.pool.root.1      AOD.31407878._000696.pool.root.1      AOD.31407878._000938.pool.root.1
AOD.31407878._000013.pool.root.1      AOD.31407878._000332.pool.root.1      AOD.31407878._000697.pool.root.1      AOD.31407878._000939.pool.root.1
AOD.31407878._000017.pool.root.1      AOD.31407878._000333.pool.root.1      AOD.31407878._000698.pool.root.1      AOD.31407878._000940.pool.root.1
AOD.31407878._000018.pool.root.1      AOD.31407878._000334.pool.root.1      AOD.31407878._000699.pool.root.1      AOD.31407878._000941.pool.root.1
AOD.31407878._000020.pool.root.1      AOD.31407878._000335.pool.root.1      AOD.31407878._000700.pool.root.1      AOD.31407878._000942.pool.root.1
```


How does the file structure look like **before** “sorted-by-directory”

DSN:

mc16_13TeV.361106.PowhegPythia8EvtGen_AZNLOCTEQ6L
1_Zee.merge.AOD.e3601_e5984_s3126_r10201_r10210_tid17
154780_00

pos	posdiff	tapenum	ctime
335	-1	A7450800	4/26/2019 2:18
360	25	A7450800	4/26/2019 2:21
392	32	A7450800	4/26/2019 2:24
393	1	A7450800	4/26/2019 2:25
413	20	A7450800	4/26/2019 2:27
418	5	A7450800	4/26/2019 2:27
422	4	A7450800	4/26/2019 2:27
424	2	A7450800	4/26/2019 2:28
427	3	A7450800	4/26/2019 2:29
432	5	A7450800	4/26/2019 2:29
436	4	A7450800	4/26/2019 2:30
453	17	A7450800	4/26/2019 2:31
454	1	A7450800	4/26/2019 2:32
456	2	A7450800	4/26/2019 2:32

- **Before** the sorted-by-directory: It was FIFO of HPSS disk cache (not the same as dCache write pools. But, they are generally not too far off in time)
- Files are written to dCache (ctime) within reasonable time windows (less than 30 minutes) in this example shown left.
- Files in different datasets are also written within the same time window. (not shown here)
 - This results in positional gap.

How does the file structure look like **after** “sorted-by-directory” option

DSN:

data22_13p6TeV.00430183.physics_Main.merge.AOD.r14190_p5449_tid31407878_00

pos	posdiff	tapenum	ctime
1022	-1	A8553200	1/1/2023 22:14
1023	1	A8553200	1/1/2023 22:24
1024	1	A8553200	1/1/2023 22:25
1025	1	A8553200	1/1/2023 22:24
1026	1	A8553200	1/1/2023 22:25
1027	1	A8553200	1/1/2023 22:18
1028	1	A8553200	1/1/2023 22:18
1029	1	A8553200	1/1/2023 22:30
1030	1	A8553200	1/1/2023 22:23
1031	1	A8553200	1/1/2023 22:31
1032	1	A8553200	1/1/2023 22:28
...
1051	1	A8553200	1/1/2023 22:53
1252	201	A8553200	1/1/2023 22:55
1253	1	A8553200	1/1/2023 22:56
1254	1	A8553200	1/1/2023 22:56
1255	1	A8553200	1/1/2023 22:58
1256	1	A8553200	1/1/2023 22:58

- **After** the sorted-by-directory: Files are sorted by directory group within HPSS disk cache.
- “Sort” happens according to HPSS rule (daily time and water mark of disk cache)
- Files in different datasets are also written within the same time window. However, they do not mix any more except
 - Sort kicked in.
 - Files are written much later.
- Gaps are mostly
 - “1” with occasional big jump.

Conditions

- LTO7 and LTO8 are used in the analysis.
 - No LTO6 . Which are being repacked.
 - They have slightly different max TH. LTO7: 300MB/s and LTO8: 360 MB/s.
- Not all tapes are investigated.
 - LTO7: about ~90%
 - LTO8: about >95%
- Repacked tapes are ignored.
- Aggregated files are ignored.
 - Small files get aggregated, resulting in the same position or posdiff of zero.
- The first file of a dataset in a tape , posdiff=-1, is also ignored.
- The very large position difference is also ignored. Posdiff<100
 - Because they contribute differently to the throughput due to “Fast Forwarding”
- Sort-by-directory enabled sometime in **Oct 2020**
 - Before = “2020-10-01 00:00:00”
 - After = “2021-01-01 00:00:00”
- All data are stored in TimeScaleDB under PostgreSQL for analysis.

Results

Average Gaps, μ , **before** sort-by-directory

dsntype	μ	N	σ	σ / \sqrt{N}
AOD	2.595	830828	5.916	0.006
EVNT	2.862	265676	7.133	0.014
HITS	2.092	426272	5.598	0.009
RAW	2.691	2385570	7.065	0.005

Average Gaps, μ , **after** sort-by-directory

μ	N	σ	σ / \sqrt{N}
1.137	2953638	2.271	0.001
2.338	77570	5.743	0.021
1.511	1824959	3.308	0.002
1.069	4120768	1.542	0.001



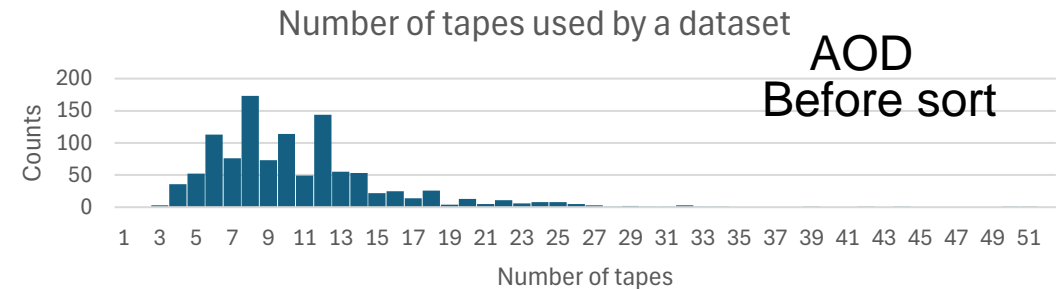
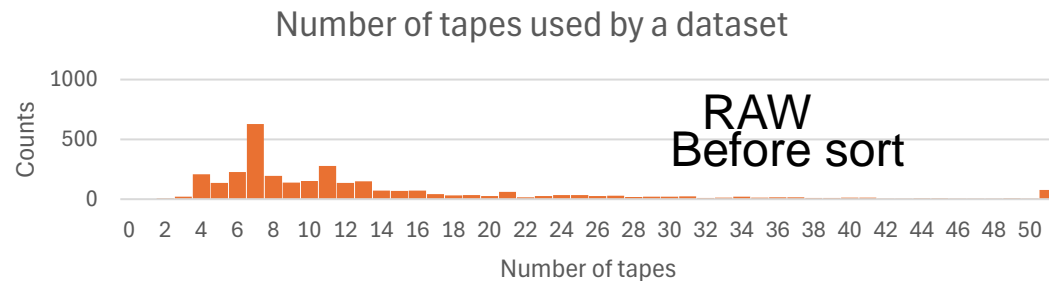
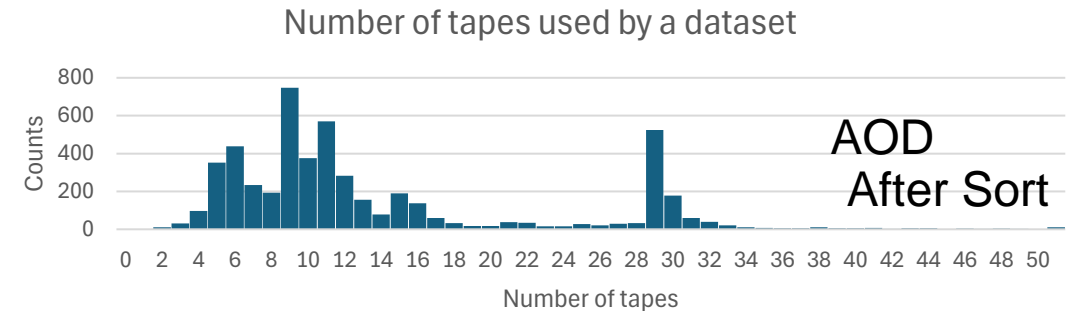
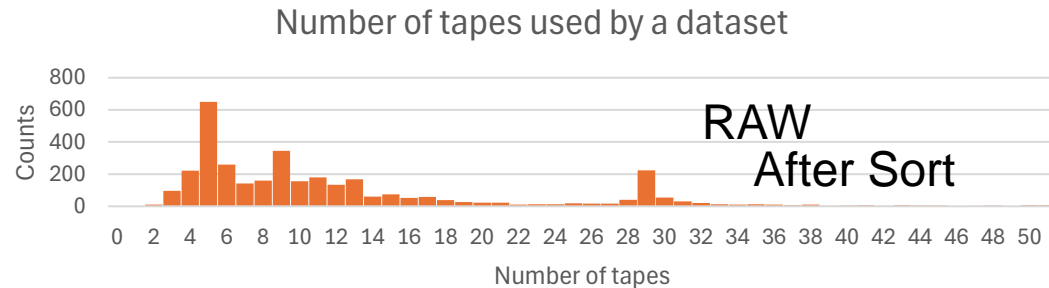
Dramatic improvement of file placements

- **AOD**: Factor of 2.3 improvement
 - If one looks at even newer files (>2023), μ is **1.05**
- **RAW**: Factor of 2.5 improvement
 - Newer files don't change the value of μ , indicating that files are coming really late.

Estimated Throughput

- AOD: Theoretical throughput / 1.13
- RAW: Theoretical throughput / 1.07

Number of tapes per dataset



- Number of tape drives for write (or read) are adjusted by HPSS admins
 - Based on the activity.
 - High inject rates might need to increase “write” drives
- Depends on the size of datasets
 - Only number of files in a dataset larger than 100 are shown

Conclusion

- Sorted write by directory has markedly improved the average file positional gap within the datasets in BNL HPSS system.
 - AOD: Before 2.6 → After 1.1 (or 1.05 for newer files)
 - RAW: Before 2.7 → After 1.1
- The dramatic decrease in average gap size should directly translate to the higher rate of the data being read from the BNL tape system.
- Sorted-by-directory is not perfect.
 - Files written much later than the others will not be in the sorted group.
 - Some files written closely in time can miss the timing of sort.
- Are we smart enough?
 - The average gap, μ , of newer files is already 1.05 .
 - Remember that the best one can do is 1.
 - μ 1.05 corresponds to 1,1,1..,(19 1s), 2. Or, $1/1.05 \sim 95\%$ of max rate.
 - Diminishing return on the further effort.
 - Simply using more drives will improve overall throughput almost always
 - If more files are packed into limited number of tapes, that will actually decrease the overall throughput of a dataset staging because it uses less than maximum number of available drives for staging.

