EPJ Web of Conferences **214**, 04022 (2019) https://doi.org/10.1051/epjconf/201921404022
CHEP 2018

Best Practices in Accessing Tape-Resident Data in HPSS*

David Yu[1*], *Guangwei Che*[1], *Tim Chou*[1] and *Ognian Novakov*[1]

[1]Scientific Data & Computing Center, Brookhaven National Laboratory, USA

Abstract. Tape is an excellent choice for archival storage because of the capacity, cost per GB and long retention intervals, but its main drawback is the slow access time due to the nature of sequential medium. Modern enterprise tape drives now support Recommended Access Ordering (RAO), which is designed to reduce data recall/retrieval times. BNL SDCC's mass storage system currently holds more than 100 PB of data on tapes, managed by HPSS. Starting with HPSS version 7.5.1, a new feature called "Tape Order Recall (TOR) has been introduced. It supports both RAO and non-RAO drives. The file access performance can be increased by 30% to 60% over the random file access. Prior to HPSS 7.5.1, we have been using an in-house developed scheduling software, aka ERADAT. ERADAT accesses files based on the file logical position order. It has demonstrated a great performance over the past decade long usage in BNL. In this paper we will present a series of test results, compare TOR and ERADAT's performance under different configurations to show how effective TOR (RAO) and ERADAT perform and what is the best solution in data recall from SDCC's tape storage.

1 Effectively Using Tape Technology

The Tape Storage System at Brookhaven National Laboratory [1] Scientific Data and Computing Center (SDCC) [2] provides it's service to the scientific experiments at RHIC[3] and LHC[4] (CERN, Geneva). The amount of our science experiments' data has increased rapidly and we currently have about 150 PB of data in our tape storage. We have put a great amount of effort into how data is saved onto tapes and how to optimize data mining and data production work flows, from a production account perspective, taking into account the time sequence and ordering of files on tape.

As the amount of our scientific experiments data is projected to be increasing very rapidly, the requirement of using tape storage is also becoming more challenging. Therefore we need to examine the architectural design of our tape storage to optimize the data archival system.

* Corresponding author: david.yu@bnl.gov (or david.yu@yahoo.com)

We have a long history of using LTO[5] technology, but other new drive and medias are worth to re-evaluating as well. Especially the Recommended Access Order (RAO) is one of the new features we need to evaluate.

1.1 Environment

1.1.1 Software

BNL's archival storage system is managed by hierarchical storage management (HSM) software called HPSS[6]. HPSS is a software developed by IBM in collaboration with various DOE National Labs [7].

In addition to HPSS, we also have a scheduler software, called Efficient Retrieval and Access to Data Archived on Tape[8] (ERADAT). ERADAT is a file retrieval scheduler, built with HPSS API calls. ERADAT is the interface between the user and HPSS. [8].

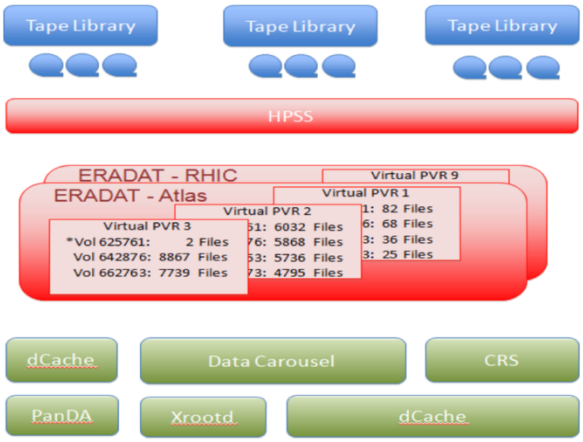

Fig 1 The relationship between ERADAT and other software.

As demonstrated in Fig 1 ERADAT aggregates all the staging requests by tape cartridge, and then sorts the requests by file logical offset, so that all the requests on the same tape can be read at once, hence to reduce redundant tape mounts. ERADAT is using linear offset ordering to access data, since this is the how LTO drive accesses the tape data.

Our goal is to find the best method that is suitable for our production environment. We tried to simulate the production system as much as possible, so we are evaluating the tape technologies by using HPSS, with ERADAT as the interface. All test data were taken from our production data.

1.1.2 Hardware

We have evaluated 3 different kind of tape technologies:

- LTO-7, T10K-D and TS1150+JD media.
- We use similar disk buffer for all 3 tape-technologies.

1.1.3 Test Environment

- HPSS core and gateway server: RHEL6.9, HPSS7.5.1p2u1, DB2v10.5p8
- HPSS mover and pftp client server: RHEL6.9, HPSS7.5.1p2u1
- Tape Libraries: Oracle StorageTek SL8500 Modular Library System, IBM TS4500 Single Frame Library.
- Tapes drives: IBM LTO-7, Oracle T10KD, IBM TS1150
- Tapes: LTO-7, Oracle T10KD, IBM JD Media.

1.2 Test case and configuration

1.2.1 *Terms and settings*

RAO: Recommended Access Order (RAO), a new feature supported by enterprise drives like T10K-D and TS11 series drives. When accessing enterprise level tapes with TOR turned on, HPSS stages files based on RAO.

Offset order: accessing data based on logical offset on the tape, sequentially.

HPSS Staging modes:

Default
by default TOR (Tape Ordered Recall) is enabled, HPSS will use the enterprise drive built in feature RAO (if available from the drive) or make linear offset ordering (if RAO not available), to schedule and submit the staging requests.

Noschedule
TOR is disabled, staging request will not be ordered before submitting to HPSS.

Staging coverage:

100 % stage
All files on the tape will be recalled.

50% stage
Half of the files on the tape will be recalled with even spacing method, i.e. retrieve every other file.

10% stage
10% files on the tape will be recalled with even spacing method.

We believe even spacing will give us a good reference baseline. It could also imply the worst situation. Fig 2 illustrated a tape with 2 wraps, total 40 files. We want to stage only 10% of the files from this tape, the 4 sample files were 10 files apart from each other.

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
40	39	38	37	36	35	34	33	32	31	30	29	28	27	26	25	24	23	22	21

Fig 2 Representing a tape with 2 wraps.

Besides the even spacing, we also did a few tests with random spacing.
Recall request submission:

ERADAT recall
ERADAT tool is used to sort the recall requests based on file's logical position order before submitting jobs to HPSS for staging. Batch submission size is 3, i.e. 3 files are concurrently submitted to HPSS for staging. TOR is disabled, i.e. Job is run under noschedule mode.

1.2.2 *Test Data*

We used 3 sets of test data: Large file, small file, and small file with aggregation.

Large Files
10 GB per file, none aggregated. A dedicated tape was fully written with the same 10G file.

Aggregated Small Files
1 GB file migrated to a dedicated tape with aggregation policy set as following: Max. file size: 2GB, Max number of files in aggregate: 30

Non-aggregated Small Files
1 GB file migrated to a dedicated tape without aggregation set.

2 Test Methods

All tests were timed for staging only: files read from tape and written on disk, no network data transfer is involved.

Submit 10% file list, 50% file list, and 100% separately to HPSS via ERADAT.

2.1 TOR disabled

Submit 3 requests to HPSS. When a file is completed, submit another one. **Error! Reference source not found.** illustrate the file submission logic. When file A is completed, file B will be activated immediately since B is already queued in HPSS. While B is being read from tape, file C will be queued in HPSS. **Error! Reference source not found.** is quoted by "Efficient Access to Massive Amounts of Tape-Resident Data" [9]

Typically, 2 threads should be enough. Since LTO-7 and TS1150 are much faster than LTO-6, we increased the queue depth to 3. Fig 3 illustrated a 3 threads buffer, while File A is being staged; File C is waiting in queued. As soon as File A is finished, File C will be the next one in the line.

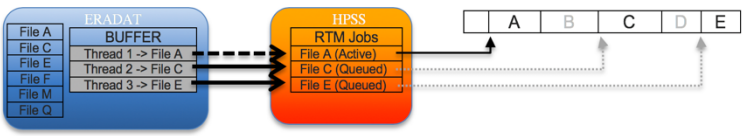

Fig 3 Buffer queue

In Fig 4, while File A is retuned, File C should be already being staging, thread 1 should be updating the status for File A, and then submit File F. With LTO-7, 300 MB/s, we use a 3

threads model, to reduce latency. A recommendation is to not over allocate the buffer as it will not be helpful but wasting thread's resources.

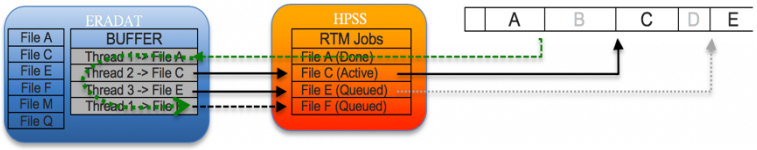

Fig 4 Buffer queue (Return)

2.2 TOR Enabled

Submit 70 bulk requests to HPSS. When a file is completed, a new file will be submitted to keep 70 requests full filled per tape in HPSS.

3 Test Results

3.1 LTO-7

3.1.1 Large File - 10 GB

When staging large files, we observed some performance improvement with TOR. However, the gain became less visible when staging more files from tape. Table 1 shows the performance difference when using TOR.

Table 1 TOR effect for 10 GB files

	No TOR	TOR	Gain %
10% 59 files	103.9	108.98	4.89%
50% 294 files	156.97	160.29	2.12%
100% 587 files	251.87	251.74	-0.05%

3.1.2 Small File 1 GB

When staging small files, we observed large performance decrease with TOR. See Table 2.

Table 2 TOR effect for Small 1 GB none-aggregated files

	No TOR	TOR	Gain %
10% 400 files	32.33	25.05	-22.52%
50% 2000 files	120.24	48.74	-59.46%
100% 4000 files	236.03	133.22	-43.56%

3.1.3 Small File 1 GB, Aggregated

When staging small files, we observed performance decrease with TOR. See Table 3.

Table 3 TOR effect for Small 1 GB Aggregated files

	No TOR	TOR	Gain %
10% 400 files	35.44	34.95	-1.38%
50% 2000 files	136.35	114.73	-15.86%
100% 4000 files	252.19	241.2	-4.36%

3.2 IBM TS1150 with 3592 JD media

3.2.1 Large File - 10 GB

We observed significant performance gain when staging large files. However the performance gain became less and less as the number of files increased. Table 4 shows near 60% gain when staging 10% files.

Table 4 RAO effects for 10 GB files

	No RAO	RAO	Gain %
10% 80 files	153.35	244.61	59.51%
50% 400 files	242.47	311.2	28.35%
100% 800 files	352.84	332.39	-5.80%

3.2.2 Small File - 1 GB

We also observed performance gain when staging smaller files from 10% coverage. However the performance gain became less and less as the number of files increased. Table 5 shows near 43.26% gain when staging 10% files, but became negative when staging more than 50% files back.

Table 5 RAO effects for small 1 GB none-aggregated files

	No RAO	RAO	Gain%
10% 980 files	73.69	105.57	43.26%
50% 5811 files	177.04	174.53	-1.42%
100% 9793 files	343.98	334.72	-2.69%

3.2.3 Small File - 1 GB Aggregated

It's a surprise to see huge performance drop when using RAO to stage aggregated files in HPSS. **Table 6** shows the poor performance in all scenarios.

Table 6 RAO effects for small 1 GB Aggregated files

	No RAO	RAO	Gain %
10% 979 files	39.82	20.18	-49.32%
50% 4899 files	177.8	56.48	-68.23%
100% 9798 files	345.61	132.08	-61.78%

4 Random Spacing

We also tested random spacing staging case. Source files list were scrambled, and randomly chosen 10% of the files with Linux shuf command. For example: shuf -n 80 file_list.

As expected, random spacing gives much better performance with both RAO on and RAO off. See Table 7 and Table 8 for details.

Table 7 Random Spacing, RAO effects for 10 GB files

Big file (10GB)	No RAO	RAO	
10% 80 files	170.81	237.72	39.17%
50% 400 files	254.22	309.72	21.83%

Table 8 Random Spacing, RAO effects for 1 GB none aggregated files

Small file (1GB)	No RAO	RAO	
10% 980 files	73.21	134.18	83.28%

As expected, RAO performed very poorly with HPSS small file aggregation, compare to linear offset ordered, showing in Table 9 below.

Table 9 Random Spacing, RAO effects for 1 GB Aggregated files

Small file Aggregated	No RAO	RAO	
10% 979 files	48.58	28.59	-41.15%

5 Conclusions

5.1 RAO and HPSS small file-aggregated block

We have discussed the small file aggregation problem with IBM HPSS team, we believe the problem was caused by the auto-seek when RAO was turned on. When auto-seek is enabled, the read head automatically moved to next block as soon as a file is read, and HPSS would have to pull it back to the beginning of the block and seek for next file.

Fig 5 Illustrated a 25% even spaced sample. We need to stage files marked yellow in color. Green represented a single aggregated data block. As the auto-seek is in effect, the read head automatically jumped to the beginning of next block.

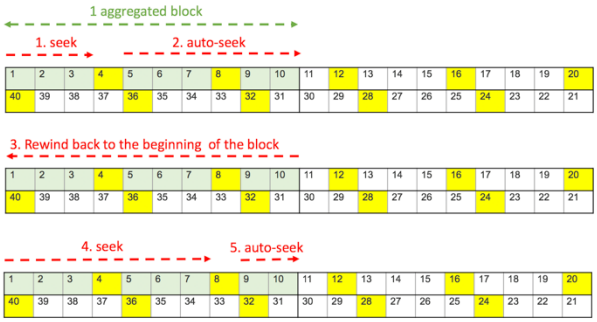

Fig 5 Impacts to HPSS small file aggregated block when using RAO with auto-seek turned on

According to IBM, they claim this bug will be fixed in next release.

When auto-seek is off, the read-head could just seek to next position from where it was left.

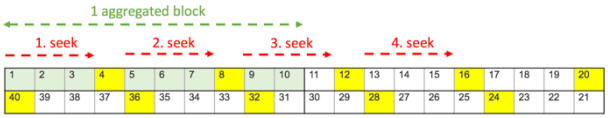

Fig 6 If auto-seek is off, we expect to see a better performance in small file aggregated block.

5.2 Best Staging Practice

RAO may be a short term solution for enterprise drives like IBM TS11xx. It works great when staging small amount of data. The best long term solution of using tape is to stage bulk amount of data, stage 50%, or more per tape-mount. The enterprise media costs more than LTO in dollar per TB, but the performance gain seem to be very positive. However, since we are a LTO user, we still cannot take any advantage from RAO. Our best staging practice is to increase the bulk requests, in order to get more performance gain.

References

1. Brookhaven National Laboratory (BNL) – http://www.bnl.gov/
2. BNL Scientific Data and Computing Center (SDCC) - https://www.bnl.gov/compsci/SDCC
3. The Relativistic Heavy Ion Collider (RHIC) - https://www.bnl.gov/RHIC/
4. The Large Hadron Collider (LHC) - https://home.cern/topics/large-hadron-collider
5. LTO - https://www.lto.org/technology/what-is-lto-technology/
6. High Performance Storage System (HPSS) - http://www.hpss-collaboration.org
7. HPSS and DOE National Labs - https://en.wikipedia.org/wiki/High_Performance_Storage_System
8. D. Yu, J. Lauret, J. Phys.: Conf. Ser. **331** 042045 (2011)
9. D. Yu, J. Lauret, J. Phys.: Conf. Ser. **898** 082024 (2017)