CTA Project Technical Note

Evaluating the Fermilab CTA Tape Storage System for Small Files

Tammy Walton
twalton@fnal.gov

June 2024

Fermilab is pursuing a more efficient tape storage archival system for managing the preservation and retrieval of experimental data. This new tape storage infrastructure employs the CERN Tape Archive (CTA) software to delegate the physical movements of tapes. Traditionally, the experimental data, which are archived to tape, are stored in files with sizes ranging from kilobytes to gigabytes. The files with sizes ≤ 250 megabytes (small files) are managed by the Small File Aggregation (SFA) software, which aggregates, stores, and retrieves them. This document presents the CTA performance for managing the small files, where the testbed CTA deployment excludes SFA. As a result, the CTA tape storage system is unproductive for files having sizes ≤ 50 megabytes.

Table of Contents

1 Introduction

Each Fermilab experiment supports distinct policies and procedures for storing data in a file. The amount of data per file can depend on various conditions, such as the environment of the experiment, the rate and scale of the electronic signals registered by the detector readout system, and the data selection algorithms implemented in the analysis workflows. Because of these diverse elements, an experiment can create scientific datasets consisting of files varying in sizes, which are able to range from kilobytes (KB) to gigabytes (GB). Regardless of the file size, all irreplaceable and long-term files must be archived to tape. This requirement enables a scientific program to efficiently manage the data, singlehandedly analyze large data volumes, and most importantly, protect the integrity of the data.

The current Fermilab tape storage system, ENSTORE [1] [2] provides the services for storing and accessing data on tape. ENSTORE employs the small file aggregration (SFA) system to manage the files with sizes ≤ 250 megabytes. SFA [3] is responsible for packaging the small files into a larger file that is suitable for tape storage, as well as retrieving the small files from tape. Between May 11th and June 10th of 2024, SFA archives ≥ 40,000 files per day. This demonstrates the importance of SFA, which allows the curent experiments to safely and efficiently transfer and recall small files to and from the mass tape storage system.

In 2025, Fermilab will replace ENSTORE with the CERN Tape Archive [4] (CTA) tape storage system. CTA operates exclusively as a tape archival system, and it commuicates with the tape hardware. For the CTA deployment at Fermilab, CTA will serve as a backend to the dCache [5] disk system, which will continue to administer the disk funtionalities. Figure 1 shows the proposed configurations for operating CTA. Currently, CTA do not support any dedicated features to handle small files for tape storage. Since the Fermilab experiments are actively producing small files and there are > 90 petabytes of data corresponding to small files on tape, it is essential to understand the CTA performance for managing small files before transitioning to the CTA tape storage system.

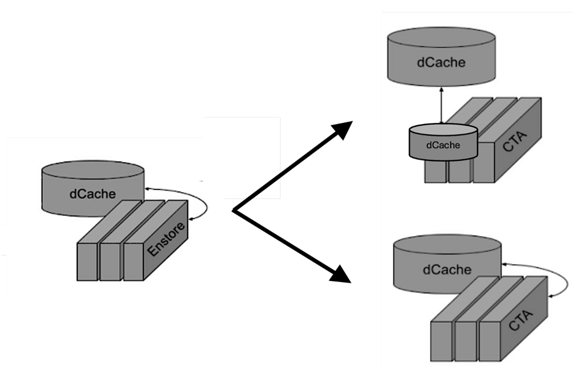


Figure 1: The current (left) and proposed (right) tape storage systems for the Fermilab experiments. The dCache disk distributes all disk functionalities and ENSTORE/CTA manages the tape storage system. For the proposed system (top-right), the small dCache solid-state disk delegates the staging of files to and from tape. *Credit: Ren Bauer*

This document presents the procedure and results given by evaluating the CTA performance for archiving and retreiving files with sizes ≤ 100 megabytes. The document is organized as the following :

- Section 2 presents a brief overview of the deployed tape storage software stack.

- Section 3 describes the various analysis workflows and results.
- Section 4 provides the summary.

2 Description of the CTA Deployment

As mentioned above, this section provides a brief overview of the implemented CTA tape storage system that is evaluated for the analysis. Figure 2 presents an illustration of the CTA deployment, where the multi-protocol, open source distributed disk storage system, EOS [6] delegates all disk functionalities. This configuration is commonly known as EOSCTA, where an instance is deployed on a Fermilab SL7 virtual machine (*storagedev201*) hosting a multi-node Kubernetes cluster. As shown in Figure 2, the CTA framework accommodates a user interface, the CTA frontend, to communicate with the EOS disk storage system, as well as delivering the tape operations using the *cta-admin* client (version *4.10.0-2*). Along with the frontend, the tape servers (*cta-taped*) are another essential feature of the CTA software architecture. The following paragraphs give a brief description of these two primary components of the CTA infrastructure.

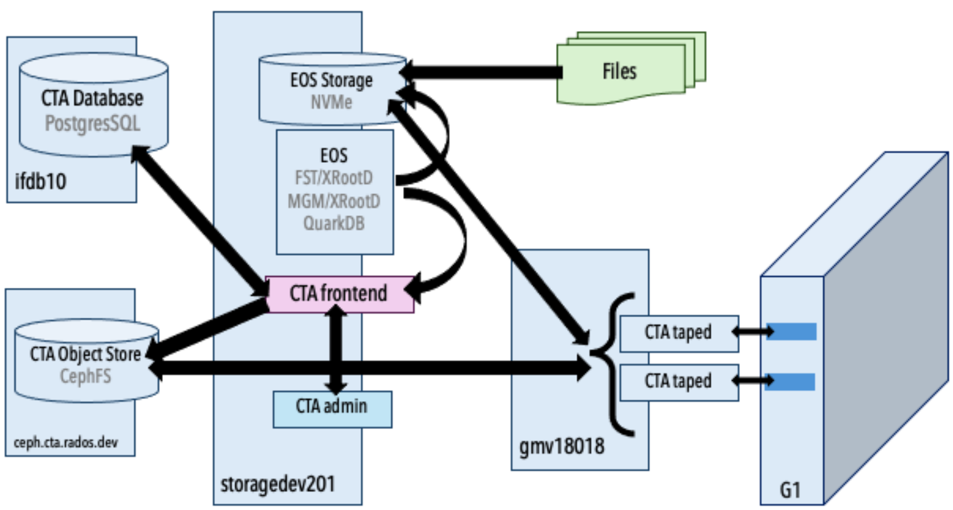


Figure 2: An illustration of the testbed CTA deployment. For this setup, EOS serves as the user interface to the disk storage system, while CTA commands the tape hardware.
Credits: Ren Bauer and Eric Vaandering

Previously stated, the CTA frontend interacts with the EOS disk system, which supports an administrative and user command line interface (CIL) along with a filesystem. Using the CLI, a user transports data to the EOS storage area (*/eos*), which is mounted on the filesystem. The stored data are managed by the EOS storage server (FST), while the EOS Metadata Generation Mechanism (MGM) service tool oversees its metadata. Both the FST and MGM daemons employ the XRootD data server. Therefore, the EOS MGM server can deliver metadata, which carries requests such as archive, retrieve, delete, and cancel to the CTA frontend. This testbed CTA frontend also utilizes the XRootD protocol and dispatches the requests to two types of storages: (i.) a queue object store that is mounted on the Chep file system and (ii.) a PostgreSQL database sending and receiving messages to and from the CTA frontend. The PostgreSQL database is maintained by the Fermilab database group. Therefore, it follows that the database is hosted on a different virtual machine (*ifdb10*). There is a one way communication between the CTA frontend and CTA object store. However, the CTA object store interacts with the tape daemons. As soon as the CTA object store accumulates a predefined number of requests, the tape servers supervise the movement of data to or from tape.

One of the responsibilities of the tape server(s) is to send a message to the EOS FST server, which writes the data to tape or stages the data to disk. Meanwhile, the CTA tape server(s)[1] delegates commands to the tape driver(s) in the tape libraries. For this EOSCTA deployment, the tape server (*cta-taped*) is deployed on a bare metal virtual machine (*gmv18018*) having direct access to the tape libraries. Unlike the traditional tape storage system, there is a one-to-one direct communication between a CTA tape server and tape driver. This functionality delivers a more efficient tape storage system. Nevertheless, the archival and retrieval degree of performance depends heavily on the CTA interaction with the tape hardware, as well as the type of tape hardware that is deployed.

2.1 Brief Overview of the Tape Hardware

The table below provides the tape information that is used for the presented study, where the tape library with the identification *TS4500G1* is employed. There are two IBM tape drivers, *LTO8D0* and *LTO8D1* and four available *LTO-8* tapes with each having 9.0 terabytes (TB) capacity.

Vid	Last Write Drive	Write Mounts	Last Read Drive	Read Mounts	Occupancy
VR5843	LTO8D0	4409	LTO8D0	28	209.6 GB
VR5871	LTO8D1	3547	LTO8D1	57	9.0 TB
VR5877	LTO8D1	6710	LTO8D0	1	208.2 MB
VR8340	LTO8D1	4125	LTO8D0	10	7.7 TB

Below is a list of the commands which the tape deamon sends to the tape drive:

- Cleaningup
- Drainingtodisk
- Mounting
- Starting
- Transfering
- Unloading
- Unmounting

[1] Tape server and tape daemon are interchangeable.

3 Evalulating Small Files

The evaluation of EOSCTA performance for archiving and retrieving small files is carried out using two types of datasets, which are named the single-binned dataset (Section 3.1) and the NOvA freight train dataset (Section 3.4). Each of these small file datasets consists of generic data that are generated using the *dd* linux command. The Section 3.1 and Section 3.4 explain the distribution of the files in each dataset.

As illustrated in Figure 3, the production of small files is conducted on the DUNE general purpose virtual machines (*dunegpvmXY*) and the files are stored in the Fermilab dCache scratch area (*/pnfs/dune/scratch/user/*). After the data for each dataset is fully generated, the data are transferred to the EOS storage area (*storagedev201*)[2]. As explained above, the EOS CLI feature enables a user to interact with the EOS software. Therefore, the EOS client and software dependencies are installed on the user virtual machine (*fermicloud*). In addition to, a SSS authenication keytab is generated which grants access. Note that the keytab is also added to the *storagedev201* machine[3]. To connect with the desired instance of EOS (*eosdev*), the following environment variables must be defined.

- export EOS_MGM_URL=root://storagedev201.fnal.gov
- export XrdSecPROTOCOL=sss
- export XrdSecSSSKT=cta.<username>.keytab

Since the EOS client is deployed on a *fermicloud* virtual machine, the user dCache scratch area must be mounted on the same machine, which is done using *sshfs*. Once the setup is completed and the permissions are given, the user can transfer data from the dCache disk to the EOS disk.

The data transfer is executed using *eos cp*. Ideally, all available tapes are in the *ACTIVE* state throughout the copying. Concurrently, the EOS MGM server is delivering requests to the CTA frontend. Once a predefined request is obtained, the CTA tape daemons will dispatch commands to the tape hardware to load the available tapes. Note that the predefined request takes into account: (i.) accumulating 100 MB of data stored on the EOS disk or (ii.) surpassing 5 minutes since the first request is detected in the CTA object store. After the tapes are mounted, the EOS FST server will write the data to tape, while the EOS MGM server updates the metadata.

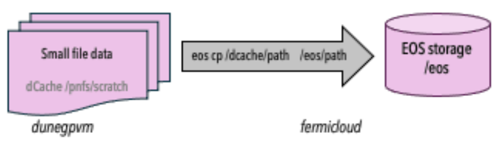


Figure 3. A diagram of data transportation from the user disk storage area to the EOS storage area.

To retrieve the data from tape, the stagging strategy embodies the following: (i.) remove the files from disk and (ii.) prepare the files for stagging. Recall that the EOS servers and the CTA frontend employ XRootD, and therefore the *xrdfs prepare* command is selected to communicate the stagging tasks. Then, the CTA tape server is responsible for retrieving the tapes. Meanwhile, the EOS FST server oversees the retrieval of data from tape and to the EOS

[2] An administrator with storagedev201 root access creates the directory and tapepool in EOS storage area.

[3] An administrator with storagedev201 root access must add the keytab to the permissions file.

disk storage area. A detailed overview of the implemented archival and retrieval procedures are presented in the remaining sections.

3.1 The Single-Binned Analysis

The single-binned dataset is composed of files with sizes spanning from 100 KB to 10 MB, where the data are generated for the following sizes:

- 100, 200, 300, 400, 500, 600, 700, 800, and 900 KB
- 1, 2, 3, 4, 5, 6, 7, 8 , 9, and 10 MB

The 19 ensembles of files are archived to tape in a specific arrangement. Starting with the 10 MB files and ending with the 100 KB files, the files undergo the archival procedure as described below:

1. The same size files are copied to a destinated same size file EOS directory using *eos cp*.
2. As previously stated, once a predefined request is reached, the data are written to tape.
3. The user monitors the CTA distribution of commands to the tape hardware.
    a. If the tape drive reports the *Cleaningup* status for an extended period and the available tapes are in the *DISABLED* state, the copying operation is terminated.
    b. A procedure is performed to bring the tape system back online (Section 3.6).
    c. Following a sucessful tape system recovery, the copying operation is restarted.
4. The user waits for a fixed number of the same size files to be archived to tape.
    a. This number is small (large) for megabytes (kilobytes) files, respectively.
5. The user records important information from the Granfana monitoring.
6. After all files are achived to tape and ≥ 30 minutes are passed, the steps 1-5 are repeated for the next-in-line data bin of same size files.

Figure 4 shows the distribution of the number of archived files for the single-binned dataset. There are 12,000 archived files for the 100 KB data bin and 4,001 archived files for the 3 MB data bin. For this evaluation of CTA, only two tapes (*VR5843* and *VR5871*) are available. Each tape consists of files from all data bins, where most of the archived data exist on the *VR5871* tape.

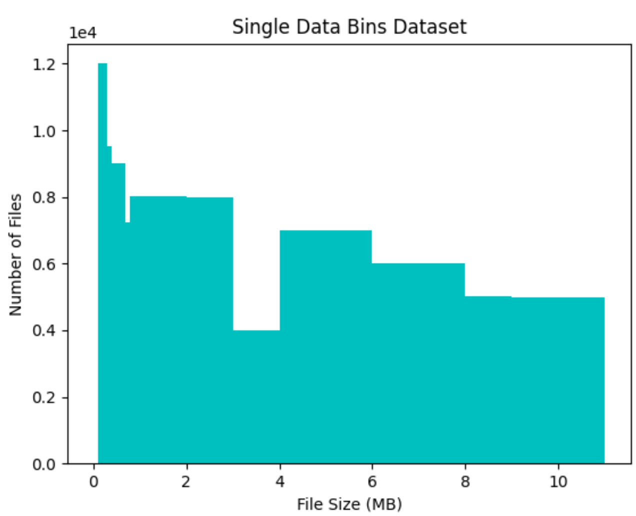


Figure 4: The distribution of the number of files for the single-binned dataset having a total size of 368 GB and 144,838 files.

Although the data are spread across multiple tapes, the data are written to one tape at a time. In addition to, the total size of the single-binned dataset is 368 GB. Therefore in theory, all data are expected to be archived on the *VR5871* tape. However, an unusual pattern is observed when archiving the data to tape, which is described below:

1. Initially, the data are written to the *VR5871* tape.
2. After an extended time period, the *LTO8D0* tape drive enters the *Cleaningup* status and the *VR5871* tape enters the *DISABLED* state.
3. Afterwards, the *VR5843* tape is mounted in the *LTO8D1* tape driver.
4. The data are written to the *VR5843* tape.
5. Following a very short time period, the *LTO8D1* tape drive enters the *Cleaningup* status.
6. Using the *cta-admin* commands, the status of the tapes and tape drivers are checked.
    a. Both tapes are declared as *DISABLED*.
    b. Both tape drives are declared as *DOWN*.
7. A recovery procedure (Section 3.6) is performed to bring the tape system back online.
    - For the majority of the data bins, only one recovery is executed.

This behavior is the most problematic for the 3 MB files and therefore, only 4,001 files are archived to tape. It is worth mentioning that the pattern is not yet understood.

Following the archival process, the retrieval of files is carried out. The retrieval techique has a similar structure as the archival procedure, which starts with the 10 MB files. The steps are the following:

1. The same size files are removed from the EOS disk, via the *xrdfs prepare -e* command.
2. The updated metadata is confirmed, via the file status (*NotOnDisk::OnTape* → *d0::t1*).
3. The same size files are prepared for stagging, via the *xrdfs prepare -s* command.
4. The stagging is monitored and important information is recorded.
5. After the stagging is completed and ≥ 60 minutes are passed, the steps 1-4 are repeated for the next-in-line data bin of the same size files.

The next section (Section 3.2) presents the results of archiving and retrieving the small files from the single-binned dataset[4].

3.2 The Results of the Single-Binned Analysis

The measurements are extracted from the data that are stored in a Prometheus database. To fully evaluate the CTA ability to archive small to very small files, all 19 ensembles of the same size files are written to tape. However, only the same size files belonging to the 3 MB to 10 MB data bins are staged to disk. Recall that two tapes are deployed. When retrieving the 3 MB files, one of the tapes (*VR5843*) malfunctions in the tape drive[5]. Therefore, the retrieval test cannot be completed for the single-binned dataset. Consequently, there are not any measurements for the 100 KB to 2 MB files, as shown in Figure 5 (blue dotted line).

[4] Note that the archival and retrieval CTA performance are provided in terms of the write and read data rates, respectively.
[5] The *VR5843* tape is recovered and active for the NOvA freight train analysis.

Previously described, after the CTA object store reports a predefined request, the CTA tape servers will send commands to the tape drives. As a result, both the CTA frontend and backend processes are running simultaneously. For the archival procedure, the CTA frontend and backend actions include: (i.) the copying of data from the dCache disk to the EOS disk and (ii.) the writing of data to tape. For the retrieval procedure, the parallel operations are: (i) preparing the data to be staged on EOS and (ii.) the recalling of data from tape. It is important to understand that both the archival and retrieval tests are not entirely executed as designed. This is due to the instability of the tape system, which emerges from attempting to manage many small files.

When the tape system stops functioning, the user is required to terminate the CTA frontend process. However, when carrying out the archival test for the 8 MB and 9 MB files (red dotted line in Figure 5), the frontend process is not killed. Therefore, all desired files are on EOS disk when the tape system is recovered. Figure 5 shows that the writing speed is faster for the 8 MB and 9 MB files. In addition to, when retrieving the 3 MB files (blue dotted line in Figure 5) from tape, the tape hardware breaks down and the CTA frontend operation is not interrupted. As a result, all desired files are prepared for stagging when the tape hardware is repaired. Figure 5 demonstrates that the reading speed is much better for the 3 MB files compared to the 4 MB to 10 MB files. Concluding, these events reveal that the data transfer speed depends heavily on the competition between the CTA frontend and backend processes.

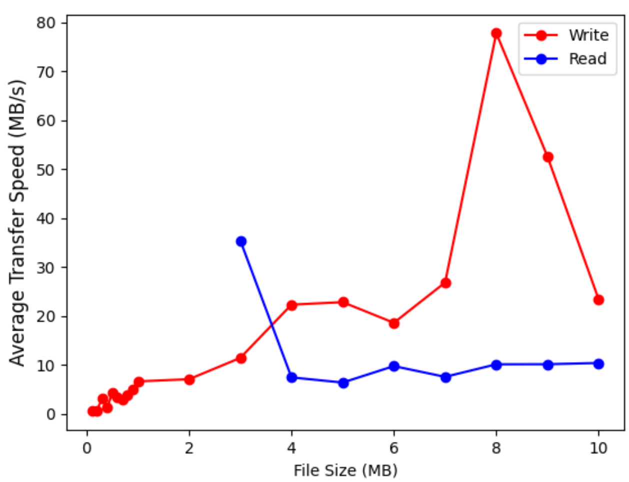


Figure 5: The data transfer speed for writing (red dotted line) and reading (blue dotted line) files with sizes ranging from 100 KB to 10 MB and 3 MB to 10 MB, respectively.

3.3 The Results of the Modified Single-Binned Analysis

The results of the single-binned analysis (Section 3.2) uncover that the CTA frontend and backend processes are encapsulated in the measurements. As explained above, this is due to parallel processing. Therefore, a modified single-binned analysis is carried out. It exercises a technique that enables the CTA frontend and backend processes to operate sequentially. Since the CTA frontend process is detached from the CTA backend process, the modified single-binned measurements capture exclusively the performance of the tape system.

The modified single-binned dataset includes files with sizes ranging from 100 KB to 10 MB, where the sizes are:

- 100 KB, 400 KB, 700 KB, 1 MB, 3 MB, 7 MB, and 10 MB

Like the original single-binned analysis, the 7 ensembles of the same size files (Figure 6) are also archived to tape in a discrete order. The archival steps are similar to the archival procedure that is described for the single-binned analysis. Starting with the 10 MB files, the steps are:

1. Change the status of each tape drive to Down.
2. Copy the same size files from the dCache disk to the EOS disk.
3. Once all data are transferred to the EOS disk, the tape drives are returned to the Up status.
4. Follow the the archival steps (2-6) that are presented for the original single-binned analysis (Section 3.1).

For the modified single-binned analysis, all four tapes (Section 2.1) are active, and the data are archived to the *VR5843, VR5877,* and *VR8340* tapes. In addition to, the tape system malfunctions only once. The disruption occurs for the 400 KB files, after > 80% of the files are archived to tape.

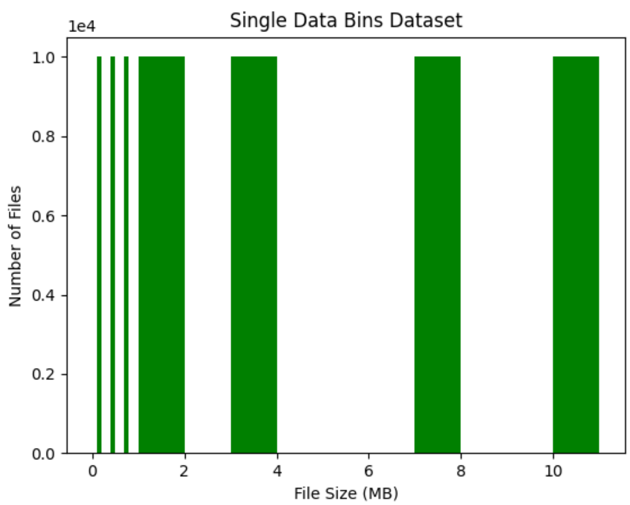


Figure 6: The distribution of the number of files for the modified single-binned dataset having a total size of 233 GB and 70,048 files.

Next, the retrieval test is performed. The methodology is the following:

1. Change the status of each tape drive to Down.
2. Follow the retrieval steps (1-4) that are outlined for the original single-binned analysis (Section 3.1).
3. Once all data are prepared for stagging, return the tape drives to the Up status.
4. After the stagging is completed, repeat steps 1-3 for the next-in-line data bin.

Bringing down the tape drives, prevents the CTA backend processes from executing while the CTA frontend processes are running. Thus, the modified archival and retrieval procedures ensure that the CTA frontend processes are fully completed before proceeding with the CTA backend processes.

IBM reports that the baseline data transfer rate is 360 MB/s for the LTO-8 tapes. As presented in Figure 7, in the case of the 10 MB files, the write (red dotted line) and read (blue dotted line) speeds reach about 28% and 20% of the ideal data transfer rate. In addition to, as the sizes of the files become smaller, the data transfer rate for writing and reading the files becomes more inadequate. Moreover, for the files with sizes < 8 MB, the write speed (red dotted line) declines faster than the read speed (blue dotted line). This is due to the inefficiencies in writing too many file marks for the given number of consecutive small files. Unfortunately, these assessments determine that CTA is unable to efficiently manage files with sizes ≤ 10 MB. However, the single-binned dataset misrepresents the conventional scientific dataset consisting of small files. Therefore, the next step is to evaluate a realistic distribution of small files.

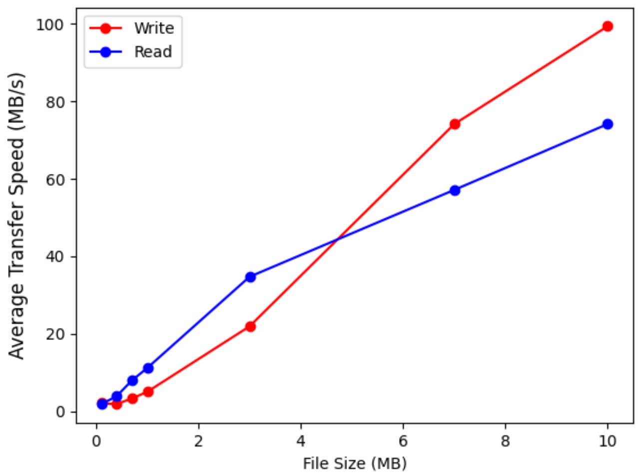


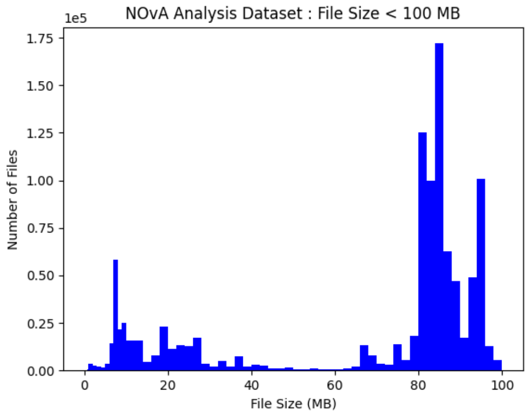
Figure 7: The write (red dotted line) and read (blue dotted line) data transfer speeds for files with sizes ranging from 100 KB to 10 MB.

3.4 Analyzing the NOvA Freight Train Dataset

An authentic experimental dataset is composed of small files that are distributed in an unstructured order. The long baseline neutrino experiment, NOvA [7] owns the largest number of small files that are archived to the ENSTORE tape storage infrastructure and continues to write a substantial number of small files to tape. This section presents the archival and retrieval tests using a NOvA practical dataset.

The NOvA freight train dataset[6] (blue histogram in Figure 8) consists of 1,088,578 files with sizes ranging from 127 KB to 5 GB. The total size of the dataset is 74.5 TB. Given that the assigned EOS storage area is 25 TB and a tape capacity is 9 TB, it is unachievable to use the NOvA freight train dataset to evaluate the performance of the testbed CTA tape storage system. Therefore, a simulated NOvA freight train dataset is employed (magenta histogram in Figure 8). Since the goal of the analysis is to assess the small files, only the files with sizes ≤ 100 MB are generated. On this account, the simulated NOvA freight train dataset consists of 123,726 generic files with sizes ranging from 100 KB to 100 MB and the dataset total size is 8.1 TB.

[6] The dataset consists of files from different data streams giving the unique structure, which resembles a freight train.



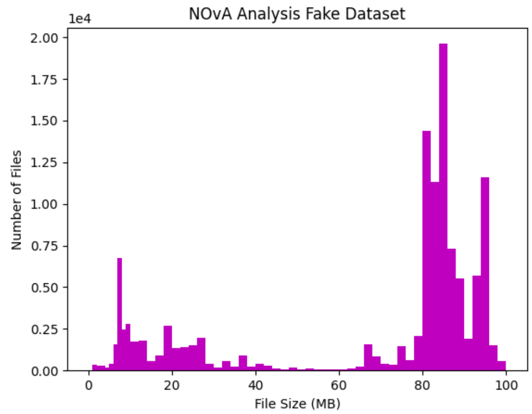


Figure 8: The distribution of the number of files for the NOvA real data (blue histogram) and simulated data (magenta histogram) datasets. The realistic dataset consists of files with sizes extending to 5 GB (not shown), where majority of the files have sizes ≤ 100 MB.

Given that an applicable NOvA-like dataset is available, the CTA performance is evaluated for a randomly order of different sized files. Unlike the previous analyses, all files are singlehandedly submitted to the archival and retrieval tests. In addition to, this deployment includes all four active tapes, which are described in Section 2.1. The procedures for archiving and retrieving the data are described below:

i. Archival Process
   1. Using *eos cp*, the 123,726 files are copied from the dCache disk to the EOS disk, where the storage area consists of 62 subdirectories and each subdirectory contains 2,000 files with sizes spanning across 100 KB to 100 MB.
   2. In parallel, the data are written to tape and CTA distribution of commands is monitored.
      - If the CTA tape system goes down, a tape recovery is performed (Section 3.6).
   3. After all files are achived, important information from the monitoring is recorded.

ii. Retrieval Process
   1. All files belonging to one subdirectory (total of 62 subdirectories) are collected.
   2. Using *xrdfs prepare -e*, the 2,000 files are evicted from the EOS disk.
   3. The metadata is checked for each file.
   4. Using *xrdfs prepare -s,* the 2,000 files are prepared for stagging.
   5. Steps 1-4 are repeated automatically for the next-in-line subdirectory.
   6. After all files are retrieved, important information from the monitoring is recorded.

Additionally, an alternative retrieval procedure, which randomly selects 25% of the files in a subdirectory, is carried out. The tape system remains accessible for all three tests. Subsequently, this evaluation of CTA completes in four days. The next section (Section 3.4) presents the results of the three sub-analyses, which comprise of archiving the data, retrieving all of the data, and retrieving 25% of the data.

3.5 The Results of the NOvA Freight Train Analysis

The measurements are extracted from the data that are recorded in the log files. Figure 9 presents the data transfer rates for writing the data to tape (blue dotted line), reading 100% of the data from tape (red dotted line), and reading 25% of the data from tape (magenta dotted line). CTA is able to efficiently write the data to tape, only for the small files with sizes ranging from 30 MB to 100 MB. Within this range, the measured write rate (blue dotted line) achieves 83% of the ideal data transfer rate. However, Figure 9 shows that there is a varying threshold to efficiently retrieve the small files from tape. In the case of retrieving all files in a directory, the read speed for the 50 MB to 100 MB files (red dotted line) is about 63% of the ideal data transfer rate. Conversely, when randomly retrieving 25% of the files in a directory, CTA performs the most productive for files with sizes ranging from 60 MB to 100 MB (magenta dotted line). Therefore, the CTA ability to retrieve small files depends heavily on the requested data and the structure of the data on tape(s).

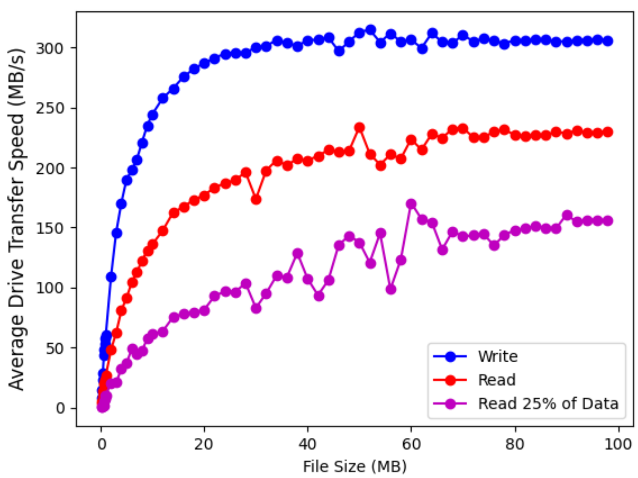


Figure 9: The data transfer rates for the three CTA performance tests, data archival (blue dotted line), data retrieval (red dotted line), and partial data retrieval (magenta dotted line).

As shown in Figure 8, the NOvA freight train dataset is comprised of a broad range of small files varying in size. Similar to the single-binned analysis, the measured write and read rates (Figure 9) illustrate that CTA is unable to manage files with sizes ≤ 10 MB. For very small files, the performance of CTA rapidly regresses. Regardless of the placement of very small files in a scientific dataset, the execution of CTA is measured to be < 20% of the potential.

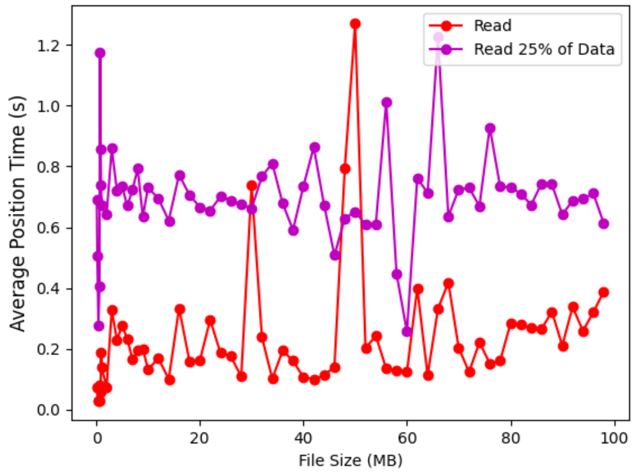


Figure 10: The tape positioning time for the NOvA freight train dataset, where the red (magenta) dotted line corresponds to retrieving 100% (25%) of the files in a directory.

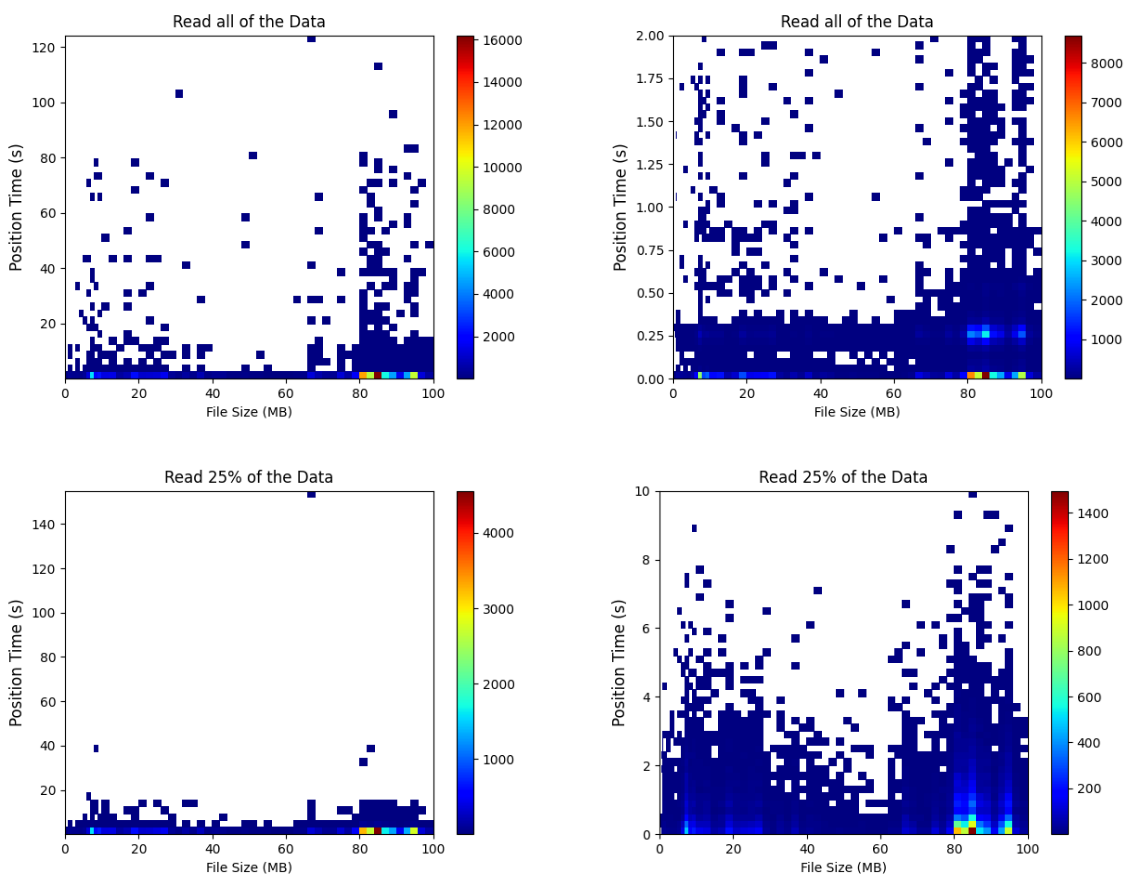


Figure 11: A two-dimension representation of the tape positioning time for the NOvA freight train dataset. The top (bottom) plots are for retrieving 100% (25%) of the files in the directory. The right plots are the zoomed in display of the left plots.

3.6 The Procedure for Recovering Tapes

As mentioned previously, the CTA tape system (*storagedev201*) did not operate successfully for the single-binned tests (Section 3.1 and Section 3.2). The tape system malfunctioning is mostly likely due to the tape drive(s) failing to eject the tape(s). Therefore, a recovery procedure (developed by Scarlet Norberg) is performed, which brings the tape system back online. The procedure is as following:

1. Go to the CTA tape server machine and check the status of tape drives, via the **cta-smc -q D**.
2. If the tape drives report the Unloaded status, run the command to eject the tapes. The commands are:
    a. **cta-smc -d -D <drive number>**
3. Go to the cta-admin client machine (*storagedev201*) and check the status of the tape drives, via the **cta-admin dr ls**.
4. If the tape drives report the Down status, bring the tape drives Up via the command:
    a. **cta-admin dr up <drive number>**
5. Check the status of the tapes, via the cta-admin tape ls.
6. If the tapes report DISABLED status, reactive the tapes via the command:
    a. **cta-admin tape ch --vid <tape id> --state active**

4 Summary

This document presents the performance of a testbed CTA tape storage system managing files with sizes ranging from 100 KB to 100 MB. The deployed CTA is evaluated for the LTO-8 tapes and uses EOS for disk functionalities. Both the single-binned and NOvA freight train analyses show that CTA is incapable of efficiently managing files with sizes ≤ 10 MB. Furthermore, the NOvA freight train analysis demonstrates that the files with sizes approaching 50 MB are unsuitable for the tape storage system. Therefore, the upcoming Fermilab CTA tape storage system must adopt the SFA service or experiments can no longer generate small files.

5 References

[1] G. Oleynik *et al*., Fermilab's Multi-Petabyte Scalable Mass Storage System, 22nd IEEE / 13th NASA Goddard Conference on Mass Storage Systems and Technologies, 2005.

[2] J. Bakken, E. Berman, C. Huang, A. Moibenko, D. Petravick and M. Zalokar, The Fermilab data storage infrastructure, 20th IEEE / 11th NASA Goddard Conference on Mass Storage Systems and Technologies, 2003.

[3] J. Hendry, A. Kulyavtsev, D. Litvintsev, A. Moibenko, S. Naymola, and G. Oleynik, Enstore Small File Aggregation User Guide, https://cd-docdb.fnal.gov/cgi-bin/sso/ShowDocument?docid=4698, 2013.

[4] M. C. Davis *et al.*, CERN Tape Archive — from development to production deployment, EPJ Web of Conferences, **214**, 04015, (2019).

[5] K. Lato and D. Litvintsev, Using Public dCache, https://cd-docdb.fnal.gov/cgi-bin/sso/ShowDocument?docid=5399, 2015.

[6] A.J. Peters, E.A. Sindrilaru, and G. Adde, EOS as the present and future solution for data storage at CERN, J. Phys.: Conf. Ser, **664**, 42042, (2015).

[7] D.S. Ayres *et al*., The NOvA Technical Design Report, FERMILAB-DESIGN-2007-01.