Examining the Impact of Data Layout on Tape on Data Recall Performance for ATLAS

Shigeki Misawa
Scientific Data and Computing Center, Brookhaven National Laboratory

Abstract:

Increases in data volumes are forcing high-energy and nuclear physics experiments to store more frequently accessed data on tape. Extracting the maximum performance from tape drives is critical to make this viable from a data availability and system cost standpoint. The nature of data ingest and retrieval in an experimental physics environment makes achieving high access performance difficult given the inherent limitations of magnetic tape. Tailoring the layout of data on tape is one key to improving read performance. This paper highlights the work in progress to characterize ATLAS data ingested in the tape system, understand how data layout, i.e. file co-location on tape and file distribution over tapes, affect read performance and how optimal data layout might be achieved in a production environment.

Problem:

- ATLAS at the Large Hadron Collider generates 10's PB/year
  - Rising to 100's PB/year starting in 2027
- Only the most active data is on disk due to cost
- Tape is increasingly being used as an active near-line store for "cooler" data
- Inefficient access to data on tape raises the cost of tape
  - Requires more tape drives
- Efficient access to data on tape is severely constrained by the inherent characteristics of tape

Limitations of Tape:

- Poor random access performance
  - Avg file to file seek time between ~ 7 sec and 30 sec [1]
- Maximum bandwidth (BW) achieved only with large streaming reads (and writes)
- Read/Write bandwidth quantized by tape drive bandwidth
  - High aggregate BW requires multiple tape drives
- Mounting a tape take ~2 minutes
- Tape libraries support a limited number of tape mounts per hour

Maximizing Drive Performance:

Effects of mount time, seek time, and data volume read/written per tape mount on tape BW

$$\% Max\ BW = \frac{1}{(1 + T_{Move}/T_{Read})}$$

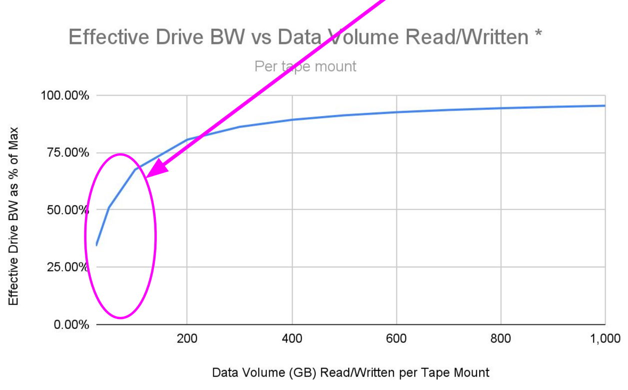
$\% Max\ BW$ = Effective BW as a fraction of the max tape drive BW

$T_{Move} = T_{Mount} + T_{Seek}$

T_{Mount} = Time to mount tape

T_{Seek} = Time seeking to data

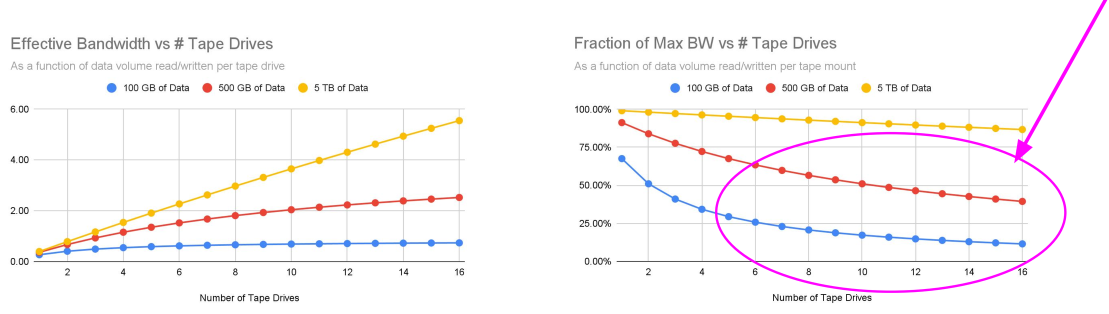
T_{Read} = Time reading/writing



* For current systems where T_{Move} ~ 120 sec, T_{Read} = Data Volume(MB) /(400 MB/sec)

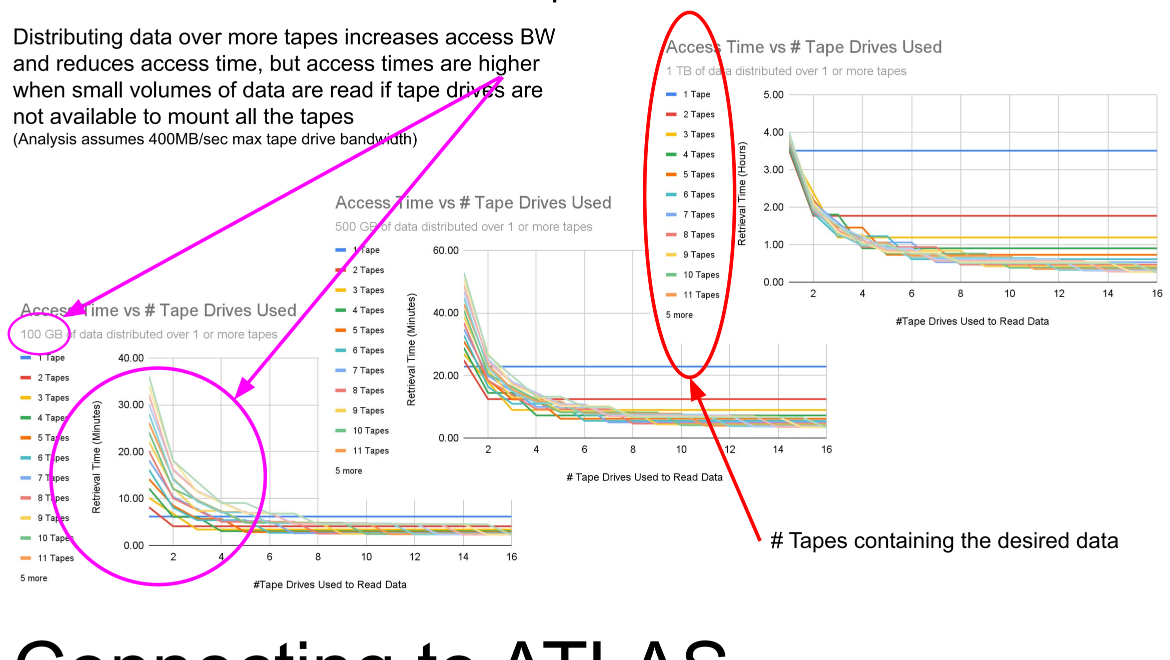
Multi-Drive Complications

More tape drives = higher bandwidth when reading/writing a given volume of data per tape mount <u>but</u> effective BW per tape drive drops

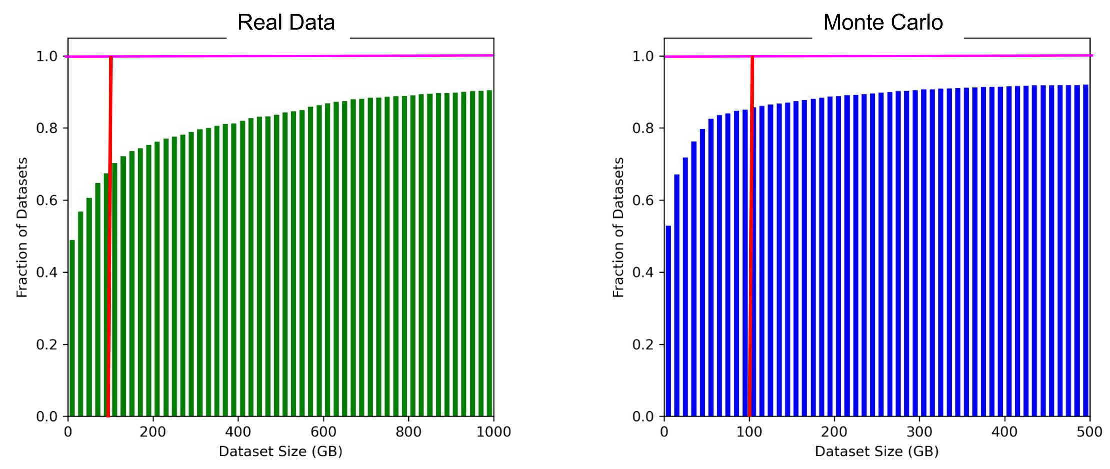


Access time is also affected if tape drives are oversubscribed

Distributing data over more tapes increases access BW and reduces access time, but access times are higher when small volumes of data are read if tape drives are not available to mount all the tapes
(Analysis assumes 400MB/sec max tape drive bandwidth)



ATLAS Dataset Size Distribution



Fraction of datasets less than a given size

From 988 DATATAPE and 8588 MCTAPE datasets written between 24/05/2022 and 14/09/2022

Consequences

Relatively small dataset size suggests low effective tape drive BW if only one dataset is read per tape mount:

- ~70% of "Real" datasets < 100GB [A]
- ~85% of Monte Carlo datasets < 100GB

Meeting ATLAS BW to tape targets requires writing multiple tapes in parallel, increasing the likelihood of a dataset being spread over multiple tapes, further reducing effective drive BW

- 10 GB/sec when data taking , > 25 tapes written in parallel [B]
- 2 GB/sec outside of data taking, > 5 tapes written in parallel [B]

[A] "Real" datasets - data from the ATLAS detector or derived from detector data, i.e., not Monte Carlo data
[B] Sustained bandwidth requirements to/from tape for ATLAS at the US ATLAS Tier 1 facility at Brookhaven

Read/Write Requests

- Requests received by tape sites are by file, not dataset.
- Read and write requests for files from disparate datasets are in the queue at any given instance
  - Tape sites see a pseudorandom sequence of file read and write requests
- By default requests are serviced by tape systems first in first out (FIFO) with the following consequences:
  - Random distribution of files in a dataset over a set of tapes
  - Intermixing of files from different datasets on tape
  - Reads with no concern for tape mounts and seeks, i.e. random access

Read Request Optimization

Simplest Optimization

- Sort read requests by tape containing the data
- Sort read request by tape order

Drawbacks

- Seeks due to intermixing of unrelated files on tape remains
- Drive inefficiencies cause by striping dataset over multiple tapes remain in cases where only one dataset is accessed per tape.

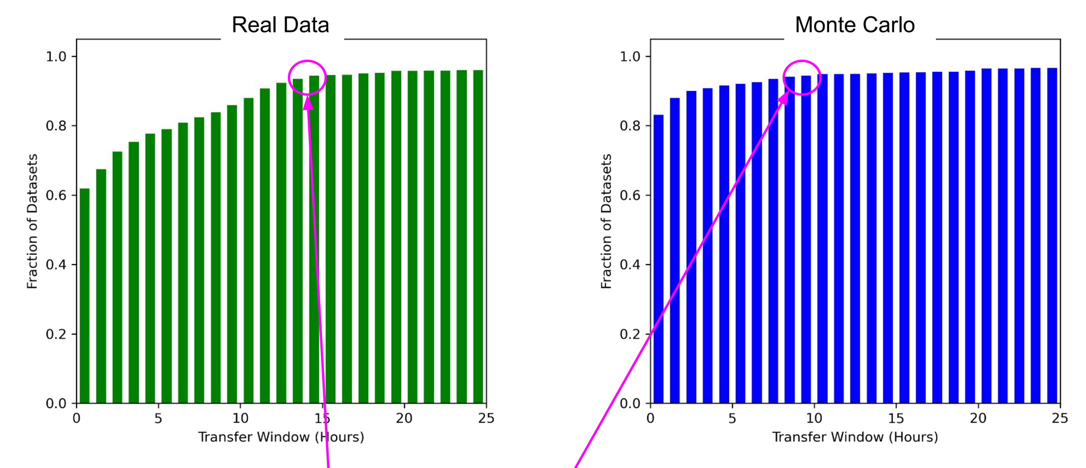
Efficiencies gained from read request optimizations limited by the layout of data on tape.

Tuning Dataset Layout

Dataset layout goals

- Reduce seek penalties through contiguous placement of files in a dataset on tape
- Increase effective drive BW by increase volume of data read per tape mount
  - Limit distribution of files in dataset to minimum number of tapes required to meet access requirements
  - Identify correlated datasets, i.e. datasets read together and co-locate them to a common set of tapes

ATLAS Dataset Transfer Time



Fraction of datasets transferred within given length of time

From 988 DATATAPE and 8588 MCTAPE datasets written between 24/05/2022 and 14/09/2022

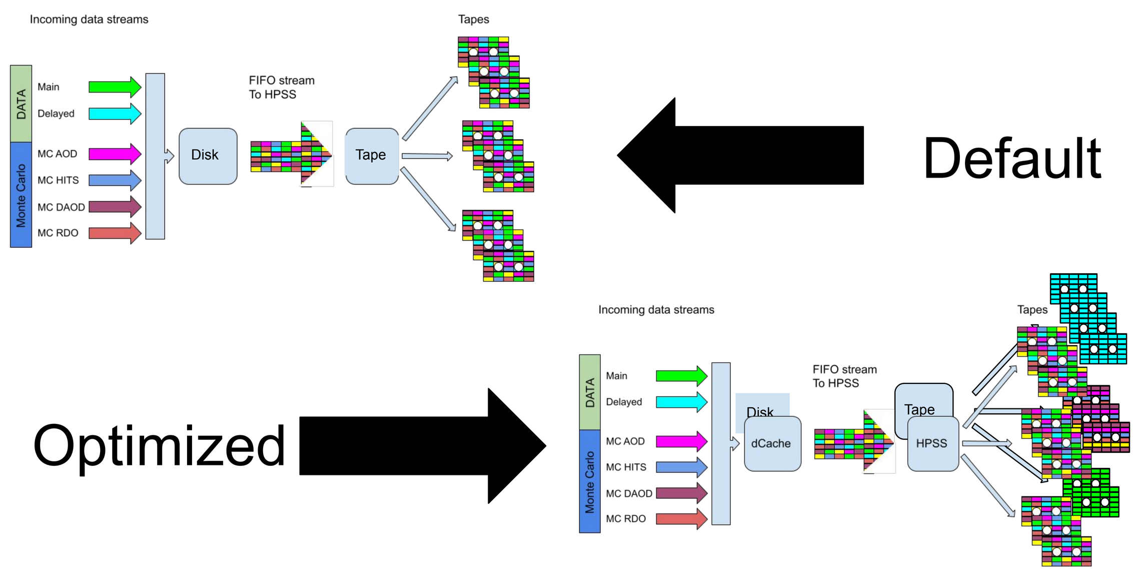
Analysis

- Rough estimate of buffering capacity needed to ensure complete datasets received before writing to tape:
  - 10GB/sec x ~15 hours = 540 TB during data taking
    - Driven by transfer of real data
  - 2 GB/sec x ~10 hours = 72 TB outside of data taking
- Better understanding of data ingest might enable reduced buffer size.
- Precise control of buffer space requires knowledge of dataset size at time of dataset transfer
- Size of buffer space affects ability to co-locate related datasets on tape. (Datasets must be received close in time)

Co-locating Multiple Datasets

- A harder problem than file in dataset co-location
- What datasets to co-locate ?
  - most obvious is real data (Specifically RAW data)
  - Input from ATLAS or analysis of historical access patterns may provide clues
- Achieving co-location is harder
  - How are candidates for co-location identified ?
  - Buffering capacity limits dataset co-location candidates to those received within a short time window
  - Dedicating specific tapes for groups of datasets a possible solution (e.g. HPSS "File Families") [3]
    - But may increase tape mounts thus impact write efficiency

Visualizing Optimal Data Layout



Conclusion

- Matching file layout on tape to recall order can potentially improve tape system file recall performance
- Co-location of files in a dataset is the most obvious and easiest to achieve
- Co-location of groups of datasets that are retrieved together is a harder problem
- Modifications of tape systems and other parts of the data distribution pipeline are necessary to implement these optimizations

Connecting to ATLAS

Basic "units" of ATLAS data are:

- File - Smallest quanta of data in the ATLAS data management system
- Dataset - Grouping of related files in the ATLAS data management system [2]. Read and write requests typically for files in a dataset

Two classes of data

1. Real data - Data generated the detector
2. Monte Carlo - Simulated data

Performance of tape systems are dependent on:

- Characteristics of ATLAS data
- How data is sent to the tape site for storage
- How data read requests are sent to the tape site for processing

Tape System Requirements

To control file layout on tape, need

- Mechanism to identify files to be grouped together on tape
- Change FIFO writing of files to tape
- Ability to steer files to selected tapes
- Disk space to buffer files

Precise file placement may not be possible with all tape systems
Size of disk space required to buffer files depends on

- Time required to receive groups of files to be written together
- Size of the file groups
- \# of file groups actively being written at a given instance

[1] LTO-CERN-HEPiX-Oct-2018, German Cancio
https://indico.cern.ch/event/730908/contributions/3153156/attachments/1732268/2800425/LTO-CERN-HEPiX-Oct-2018-germancancio.pdf
[2] Rucio - Scientific Data Management - https://rucio.cern.ch
[3] High Performance Storage System (HPSS) - https://www.hpss-collaboration.org/documents/hpss_10.2_users_guide.pdf
[4] An NFS Trace Player for File System Evaluation
https://dash.harvard.edu/handle/1/25620499

Future Work

Method is needed to evaluate the effectiveness of data layout optimizations

- Evaluate cost to deploy
- Quantify improvements in read performance

A tape system "I/O trace player" to simulate different data layout optimizations is being considered. The technique, commonly used to evaluate file systems, takes real world I/O traces (access logs) and "plays" them on the a simulation of the system being evaluated [4]. This technique eliminates the problem of creating synthetic access patterns that faithfully represent real world I/O.

U.S. DEPARTMENT OF ENERGY

Brookhaven National Laboratory
www.bnl.gov