A public dataset of magnetic tape file description and reading requests

In this document, we introduce a dataset containing the position and size of files on magnetic tapes, associated to user reading requests on these tapes from a production system. The dataset is freely accessible online using the following link: https://figshare.com/s/a77d6b2687ab69416557.

1 Context

The target computing center, from which our dataset is extracted, uses tape storage for long-term projects in the fields of Particle Physics and Astrophysics. In this context, we had access to logs of the tape system from a period of high activity. The center uses the Spectra Tfinity library, and has 48 reading engines TS1160 with 6700 Jaguar E magnetic tapes with a capacity of 20TB each.

The raw dataset covers three weeks of activities. It contains millions of lines of reading, writing and update requests with their associated timestamp. It also details positioning operations and delays of the device heads. For obvious privacy issues, we cannot make the whole raw dataset public, but only some anonymized features.

In this work, we were interested in getting a description of magnetic tapes (position and size of files on magnetic tapes), associated to user reading requests on these files. The former knowledge is accessed through description files of the tapes, given by the system. The latter is obtained from the raw logs, after several steps of filtering.

We first removed all lines from the raw dataset that do not concern reading operations. This gives us a list of 169 tapes, covering a total of $3,387,669$ files. Each tape is divided into segments containing files or *aggregates* of files. The size and number of segments depend on the tape. In a segment, the files are described by several features such as position and size. The current setup in the computing center allows to write *aggregates* of files on the tapes, *i.e.,* a batch of related files that can be written sequentially. A segment contains an aggregate if there is more than one file referenced in this segment. Within an aggregate, the position of a file is given as a couple (position,offset) here the position is actually the beginning of the aggregate, thus the beginning of a segment. Note that an aggregate can span across several segments. We discarded such aggregates and their associated requests to focus on aggregates lying on a single segment. Reading files inside an aggregate is not straightforward and generates a non-negligible overhead as the head is required to go to the start of the aggregate before reading a file. To ease the extraction of our sequences of requests, we considered that a requested file inside an aggregate will be treated as a request to read the whole aggregate. Such a behavior actually represents a strategy of buffering when aggregates are stored on disks after a file is requested within, in order to avoid the costly operations of accessing a file in aggregates. Thus, all the file requests in the same aggregate are replaced by a single request for a file of the size of this aggregate, and we associate to this file a number of requests equal to the number of files in the aggregate.

Overall, the final processing of the logs gives us 169 tapes with a total of $119,708$ files stored on it after the filtering of aggregates, according to the tape description files of the system at the considered period in the logs. The exploitation of the system logs allowed us to extract $28,853$ unique file requests on these tapes, and a total of $615,324$ user requests over these files.

This dataset is, to the best of our knowledge, the first publicly available dataset on magnetic tape storage. In the next paragraph, we describe the different files of the dataset.

2 Dataset content

Table 1 gives a brief summary of the dataset in terms of tape size and number of requests. We have a large panel of tape size, going to hundreds of files to thousands of files. The same observation stands for the number of file requests (31 to 852). Figure 1 represents the distribution of unique files requested for the different size of tapes. Even though we mostly have tapes of size less than a thousand files with a number of unique requested files less than 400, the dataset is not homogeneous and presents various amount of unique file requested on the tapes. We display in Figure 2, for each tape, the distribution of the total number of user requests with the number of unique files requested. We also observe that the total number of user requests is different between tapes having very similar number of unique files requested. Hence, we believe this dataset is heterogeneous and suitable for performance evaluation of a magnetic tape system storage.

We now describe the content of the public folder.

'list_of_tape.txt' This file lists the name of the 169 tapes in the dataset. For each tape, there is a file listing all the user requests on this tape in the folder **requests**, and a file describing the content of the tape in the folder **tapes**. The tapes are named under the format *TAPEXXX.txt* where XXX varies from 001 to 169.

requests folder For each tape, this folder contains a request file with two columns `index,nb_requests`. The former refers to the index of the requested file on the tape (see **tapes** folder) associated to the number of requests for this file. The maximum number of distinct files requested for one tape is equal to 852, and the minimum number is 31. The median value is 148 requests (for a tape of size 531), and the mean is 170 requests. Regarding the total number of user requests on one tape, the maximum is 15477 and the minimum is 1182, for a median value of 2669 files and a mean of 3640.

tapes folder This folder contains a description file of each tape in the dataset. From the left (position 0) to the left of each tape, the file describes the different segments of the tape.
It contains four columns `id,cumulative_position,segment_size,index`. The `id` column corresponds to the id number of the segment on the tape given by the system. The next two columns respectively refer to the cumulative position of the segment from the left of the tape, and its size. Finally, the column `index` is used as the id of the file on the tape starting from 1 for the leftmost file. This fourth column is used to match the `index` column of the **requests** files. The largest tape contains 4141 files, and the smallest one 111. The median size is 489 files and the mean size is 708 files.

3 Perspectives

This dataset allowed us to evaluate several algorithms on realistic data extracted from the logs of a production computing center. We expect this dataset to be a first step in the achievement of large-scale datasets of such types. Logs from a large time period can be envisioned as an extension to this dataset.

In this work, we only considered reading requests from user in the framework of the Linear Tape Scheduling Problem. However, the raw logs contains much more information that one could expect to use. Knowledge about time processing of reading operations and positioning operations performed by the multiple device heads could be leveraged to better model seeking speed and reading speed. With a fast parsing of the logs, the positioning time seems to impact the performance much more than the reading time. Hence, modeling the seeking speed of the device seems to be important in order to provide realistic cost models of the process. Temporal aspects of the raw dataset could also be exploited for a usage in online problems, for instance.

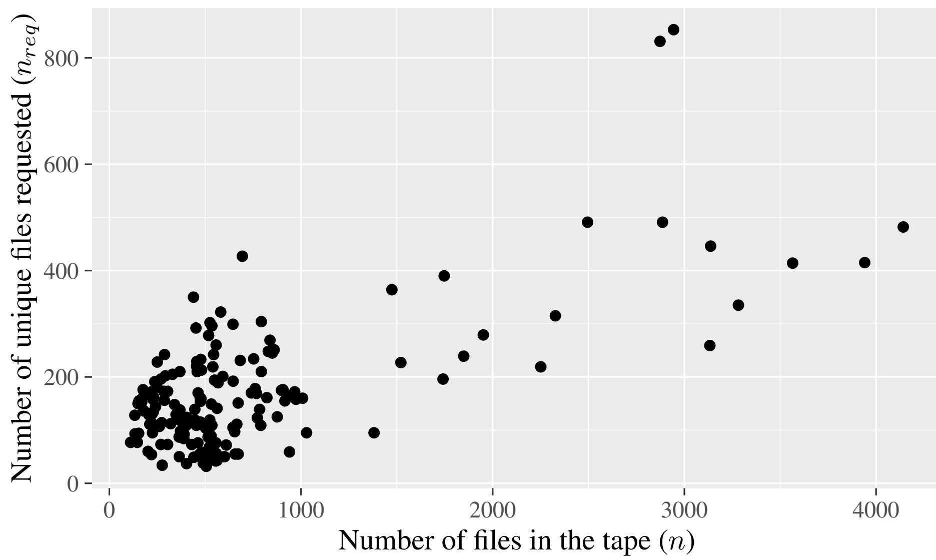

Figure 1: Illustration of the tape dataset with the number of files in each tape in function of the number of unique requested files in it.

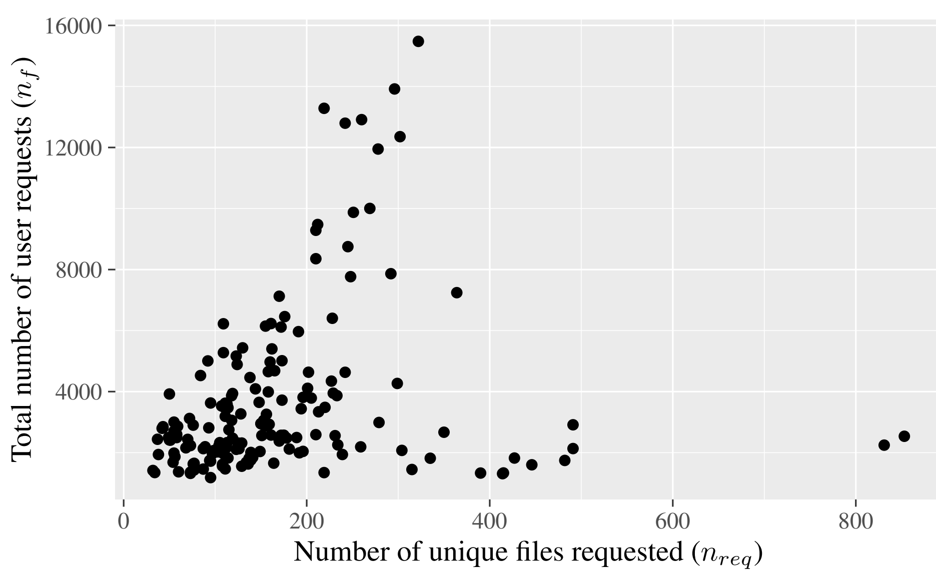

Figure 2: Illustration of the tape dataset with the number of unique requested files in each tape in function of the total number of user requests in it.

	Tape size (n_f)	# Files Requested (n_{req})	# Total User Requests (n)
Maximum	4141 (TAPE104.txt)	852 (TAPE109.txt)	15477 (TAPE139.txt)
Minimum	111 (TAPE010.txt)	31 (TAPE115.txt)	1182 (TAPE082.txt)
Median	489 (TAPE118.txt)	148 (TAPE041.txt)	2669 (TAPE078.txt)
Mean	708	170	3640

Table 1: Overview of some dataset features.