Managing data recovery for Long Term Data Preservation

Stefano Dal Pra[1,*], *Antonio* Falabella[1], *Enrico* Fattibene[1], and *Pier Paolo* Ricci[1]

[1]INFN-CNAF, viale Berti-Pichat 6/2, 40127 Bologna, Italy

Abstract. In the latest years, CNAF worked at a project of Long Term Data Preservation (LTDP) for the CDF experiment, that ran at Fermilab after 1985. A part of this project has the goal of archiving data produced during Run I into recent and reliable storage devices, in order to preserve their availability for further access through not obsolete technologies. In this paper, we report and explain the work done to manage the process of retrieving the aforementioned data, which were stored into about four thousands 2.5/5GB 8mm tape cartridges of different producers, which were widely popular in the nineties. The hardware setup for tape reading is briefly detailed. Particular focus is on describing in-house software tools and backend database that have been set up to drive and orchestrate the tape readers and to deal with the high number of possible problems arising during the process of reading data from hardly reliable media. The outcome of each operation is accounted into the database, making possible to monitor the overall progress and to retry unsuccessful read attempts at a later stage. The implemented solution has proved effective at reading a first 20% of the total amount. The process is currently ongoing. Even though a few aspects of this work are strictly dependant on how the CDF experiment organized its datasets, we believe that several decisions taken and the overall organization still make sense on a variety of use cases, where a relevant amount of data has to be retrieved from obsolete media.

1 Introduction

Starting from 2014 CNAF has been working on a Long Term Data Preservation (LTDP) [1] project, whose goal is to keep the data and the analysis software of a scientific experiment available over time, preventing them to become unusable because of hardware and software obsolescence.

One task of interest for this project is extracting data from old storage media to replicate them into modern and reliable devices in order to preserve their availability over long periods of time.

This paper focuses on the management of the recovery of the data produced during the Run I by the CDF experiment [2] from 1992 to 1997 and stored on $\sim$ 4000 8mm tapes. This operation was estimated to be feasible after having success at reading a bunch of 20 test tapes delivered to CNAF in early 2016 using old Exabyte compatible tape drives connected via SCSI to modern OS Servers (Linux S.L.6) [3].

*e-mail: stefano.dalpra@cnaf.infn.it

2 The old data tape setup

Having to deal with a large set of old cartridges to be read with an old and technologically obsoleted set of tape drives, we did a bit of investigation in an attempt of planning a work strategy that could help us to minimize operational problems such as drive failures, read errors and mandatory drive clean requests.

Initially we tried to learn more about the media tape used during those years, with particular interest about the fact that the tape cartridges came in two different brands: SONY Data type QG-112M (∼1000 tapes) and FUJI P6-120 (the remaining ∼3000).

The latter was a cheaper model widely adopted in VCR cameras, thus designed to store digital video streams rather than IT data storage. However, a comparison test [4] performed in 1990 between FUJI and SONY 8mm video tapes, simulating an accelerated aging of the two brand tape media, measured better performances (lower read error rates) for the FUJI tapes. This probably explains why the FUJI media outnumbers the SONY ones.

To begin our work we had at our disposal the following:

- A set of ∼ 4000 2.5/5GB 8mm data and video tape cartridges apparently written from 1992 to 1996.
- A set of refurbished autoloader (2× EZ17, 7–slots, 1× EXB–210, 10 slots)
- A textfiles based catalog describing tape content (write date included: this is how we know when the tapes have been written).
- A working set of scsi (`mtx` and `scsitape`) commands to perform basic operations on the autoloader and on the drive

3 The rescue problem

Our goal is to manage the extraction of most of the readable files from the old data tape set and copy them on current and reliable storage. The available autoloaders can hold up to 7 or 10 tapes at once and a script can quite easily automate the interaction with the tape drives and the loaders. This is a good time-saving respect to manually loading and ejecting tapes one at a time.

Moreover, because of the nature of the data to be read, a few failures and data loss are acceptable up to a certain extent, meaning that a "best effort" reading approach can be sufficient: in case of read failure from a tape, we can simply just take note of the event and go on reading the next one, postponing a further read attempt to a possible second phase, to only be performed in case of real need. It's also worth mentioning the fact that no checksum was available for the files, so the only possible read failure event is due to hardware level breakdown.

From previous experience we are aware that degraded magnetic tape can rapidly undermine the sensitivity of the drive head and its health status. When detecting an anomalous high rate of read errors, the drive firmware halts the device, thus requiring a mandatory cleaning operation before becoming fully operational again. This in turn requires manual intervention and conspicuous loss of time.

3.1 Managing the reading process

To organize and automate at most the process we need to write a software tool being able to independently operate all the available autoloader in parallel. We also need to collect a complete log of every single operation (tape load/eject, drive mount/umount, seek to next file position, read file header, read file and so on). These information have proved useful when troubleshooting problems from early stage of development onward.

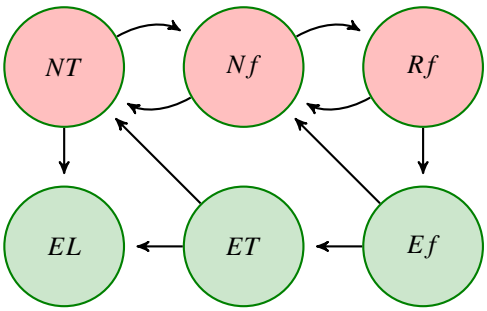

Figure 1. The Status Transition Map. The software tool operates as a Finite State Machine. In Normal Operation Mode (pink circles) the states are: **Next Tape** to umount and eject, then load and mount a new tape and recognize it by reading its header; **Next file** to retrieve from the database the name and the position of the next file in the tape; **Read file** to seek at the start position of the file, read and check for consistency its header, read the file. The error conditions (green circles) are: **Error file** an error before or while reading the file. It might be due to incoherent header or actual I/O error; **Error Tape** The tape could not be read further; **Error Loader** No more tapes could be loaded.

3.2 Tape contents and metadata

During the early stages of evaluation, the organization of data and metadata on the tape was found and verified to be consistent with the file catalog provided to us as a set of textfiles. Every tape start with a 140 bytes long tape header, containing the tape name. We refer to this as a reference to lookup the list of files to be extracted from this media. Reading forward, a 140 byte header contains the file name and size, then the actual file content, followed by a 140 byte tail. It must be noted that the same file name can occur more times in a tape. The one to be actually extracted is indicated by seek position, which is known from the text file catalog, the other ones are to be considered as deleted (unlinked) files. Because of missing CRC code in the file headers to match with the provided metadata, every successful file read is assumed to be a successful retrieval. The only viable option to validate a file once retrieved from tape is that of processing it with CDF software.

4 Software tool

To address our requirements, a Python2.7 script and a PostgreSQL backend database have been developed.

4.1 Python script and backend database

The python script operate the drives and the autoloaders to control seek-to-position and file reading. In case of blocking errors or end of the current work session an email is delivered to notify the event. Each tape is sequentially read and files are copied to a local folder named after the corresponding tape and an adler32 checksum is computed for each file. At completion, the folder is moved to online disk in a Hierarchical Storage Manager volume handled by GEMSS [5].

The backend database has been initially populated from the textfile catalog provided by CDF. These data mainly report for each tape the list of active files and their position in the media. We consider files present at the position indicated by the catalog to be *active* and the ones to be read. Other files on the tape are not considered for retrieval. On read success, filesize, reading times (start time, end time) and an adler32 checksum are stored in the database. In case of failure a file status indicator is correspondingly updated.

The overall activity is defined by a Finite State Machine describing the management of a single autoloader (Figure 1) where the next status is driven by the outcome of the current one.

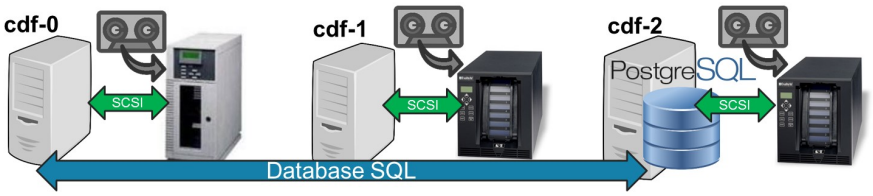

Figure 2. Each available autoloader is directly connected via SCSI to a Linux SL6 machine, running its own instance of the python script. The first machine also hosts the PostgreSQL server.

4.2 Tapes and read problems

The software tool was started in June 2017, initially operating with a single autoloader. Figure 2 represents the final setup, obtained after adding two more autoloaders in the following two weeks. The read progress over time is reported by Figure 3 and Figure 4 for the first two months of activity, during which a total of 1.35 TB over nearly 630 tapes were read, ~ 50 of which partially or completely defective. Several times, a read error had the effect of putting the drive in "cleaning request", thus requiring manual intervention, and this mostly happened when dealing with the P6-120 brand. Apparently, 27 years after the test, the video tapes prove to be more error prone. Trying to reduce the number of drive failure we decided to read the data tapes first.

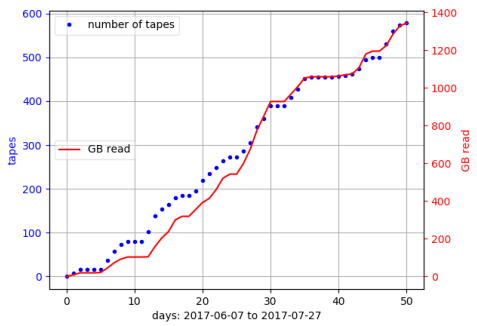

Figure 3. Number of tapes and GB read per day.

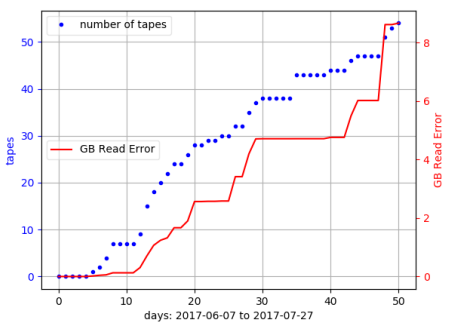

Figure 4. Number of failed tapes and GB per day.

5 Results and comments

The implemented solution has proved effective at reading a first 20% of the total amount. Table 1 reports the weekly amount of files and data read by each autoloader. It can be observed how one drive (*cdf-0*) have had discontinous performances, requiring manual intervention more frequently than the other two. Moreover, after nine weeks of operation, all the drives exhibit worse performances, requiring manual intervention more frequently.

The model presented here strictly depends on how CDF organized its own experiment datasets, however we believe that several design choices and verified solutions adopted in this case could provide useful inspiration to other use cases where data retrieval from a large number of old tape cartridges or similar sequential storage devices is involved.

Table 1. Read activity by week and drive. The three available drives have had different performances. Drive *cdf-0* for example have had more frequent failures than the other two. However, after nine weeks all three the readers have frequent blocking errors with drives requiring cleaning more frequently than expected

week	*nfiles*	*ntapes*	*GiB*	*cdf–0*	*cdf–1*	*cdf–2*
23	1207	15	19.52	0	1207	0
24	6369	62	83.84	1600	4767	2
25	5526	100	215.82	781	2620	2125
26	3891	83	223.76	18	1095	2778
27	3852	111	385.17	866	1288	1698
28	5635	63	131.36	0	2385	3250
29	1021	41	136.05	99	376	546
30	5744	77	151.28	0	840	4904
31	240	2	4.36	0	240	0

References

[1] S. Amerio, L Chiarelli, L dell'Agnello, D. De Girolamo, D. Gregori, M. Pezzi, A. Prosperini, P. P. Ricci, F. Rosso, S. Zani, in Journal of Physics: Conference Series, Vol. **513**, *Long Term Data Preservation for CDF at INFN-CNAF*, p. 042011 (2014)

[2] F. Abe, H. Akimoto, A. Akopian, MG. Albrow, SR Amendolia, D. Amidei, J. Antos, C. Anway-Wiese, S. Aota, G. Apollinari and others, in Physical review letters, Vol. **74**, *Observation of top quark production in $\bar{p}$ p collisions with the Collider Detector at Fermilab*, 2626–2631 (1995)

[3] Ricci, P P and Cavalli, A and Dal Pra, S and Falabella, A and Fattibene, E and Pezzi, M and Amerio, S, Journal of Physics: Conference Series, Vol. **1085**, in *Last developments of the INFN CNAF Long Term Data Preservation (LTDP) project: the CDF data recover and safekeeping*,032050 (2018)

[4] R. Krull, FNAL, Online, *8mm video Tape Test*, http://lss.fnal.gov/archive/test-tm/1000/fermilab-tm-1702.pdf, (1990)

[5] P. P. Ricci, D. Bonacorsi, A. Cavalli, L. Dell'Agnello, D. Gregori, A. Prosperini, L. Rinaldi, V. Sapunenko, V. Vagnoni, in Journal of Physics: Conference Series, Vol. **396**, *The Grid Enabled Mass Storage System (GEMSS): the Storage and Data management system used at the INFN Tier1 at CNAF*, 042051, (2012)