Dynamic sharing of tape drives accessing scientific data

A. Cavalli, D. Cesini, E. Fattibene, L. Morganti, A. Prosperini, P. P. Ricci and V. Sapunenko

INFN CNAF, viale Berti Pichat 6/2 40127 Bologna, Italy

E-mail: enrico.fattibene@cnaf.infn.it

Abstract. The data management infrastructure operated at CNAF, the central computing and storage facility of INFN (Italian Institute for Nuclear Physics), is based on both disk and tape storage resources. About 40 Petabytes of scientific data produced by LHC (Large Hadron Collider) at CERN and other experiments in which INFN is involved are stored on tape. This is the highest latency storage tier within HSM (Hierarchical Storage Management) environment. Writing and reading requests on tape media are satisfied through a set of Oracle-StorageTek T10000D tape drives, shared among different scientific communities. In the next years, the usage of tape drives will become more intense not only due to the growing amount of scientific data to manage but also due to general trend to use tapes as "slow disk", announced by the main user communities. In order to reduce hardware purchases, a key point is to minimize the inactivity periods of tape drives. In this paper we present a software solution designed to optimize the efficiency of the shared usage of tape drives in our environment.

1. Tape-based facility at CNAF

CNAF is the major Data Center of INFN, offering resources and services to communities involved in scientific collaborations. As INFN participates to the LHC, the largest and most powerful particle accelerator in the world, CNAF is one of the 11 Tier-1 centers of the WLCG (Worldwide LHC Computing Grid), that receive data produced by the LHC experiments (ALICE, ATLAS, CMS, LHCb). Data coming from LHC are of the order of 1GB/sec on a monthly average, with peaks of 3 GB/s or more. Moreover, CNAF Data Center provides computing and storage facilities for 30 other experiments in which INFN is involved, belonging to Astrophysics, Astro-particle Physics and High Energy Physics domains. Data are stored on both disk and tape storage resources. At the time of writing, ~20 PB of data reside on disk and ~44 PB on tape.

1.1. Infrastructure and services

CNAF mass-storage infrastructure is based on a tape library Oracle-StorageTek SL8500 equipped with 17 T10000D tape drives used for scientific data and 9 T10000C drives used only for backup and recovery service. The overall capacity of the SL8500 library is 10000 slots, so ~85 PB could be stored with the existing technology. Tape-based storage is the highest latency storage tier within a HSM (Hierarchical Space Management) environment. In order to allow data access to scientific communities, the Storage Management group operates services based on a set of software packages:

- IBM Spectrum Scale [1]: formerly GPFS (General Parallel File System), a high-performance clustered file system developed by IBM. File systems can be partitioned into a number of

storage pools implementing file placement policies and data migration rules from one pool to another according to some user-defined criteria.

- ISP (IBM Spectrum Protect) [2]: formerly TSM (Tivoli Storage Manager), a proprietary software designed by IBM, one of the leaders in data protection solutions. It offers a HSM extension to manage migrations from disk to tape and recalls from tape to disk of data hosted on Spectrum Scale file systems.
- StoRM (Storage Resource Manager) [3]: a software released by INFN based on SRM (Storage Resource Management) interface to access storage resources.
- GEMSS (Grid Enabled Mass Storage System) [4]: a software developed by INFN that provides a full HSM integration of Spectrum Scale, ISP and StoRM. It has been designed to optimize migration and recall operations.

Migrations and recalls are managed through HSM servers equipped with Fiber Channel connections to both storage disk and tape drives. Each server can be configured to handle one or more Spectrum Scale file systems. For each file system, an active HSM server is running and another standby server is configured and can be turned on in case of unavailability of the active one. At the moment, CNAF operates 6 active and 6 standby HSM servers.

1.2. GEMSS recalls

ISP HSM software can recall data from tape to disk using two possible methods: selective and transparent recalls. In case of selective recalls, the user (or a specific service on his/her behalf) asks for a file to be recalled from tape before submitting a job to a worker node, i.e. before making the first file access. This typically happens in the WLCG world, where the recall request is made via SRM commands, i. e. through StoRM service. Only when all the needed files have been recalled, the access is performed. Transparent recalls are triggered by a read operation (usually from user jobs) of a migrated file, i. e. in case only the stub file is present on disk. When the recall is finished and the file is accessible on disk, the control is given back to the user's process.

The standard ISP HSM behaviour consists in recalling files as soon as they are requested by users, following the order of the requests. As users have no knowledge of where the files are stored, and in particular of the way the files are ordered within a tape, such a procedure ends up in an inefficient usage of the tape resources. To overcome this limitation, GEMSS implements its own aggregation and reordering of tape recalls before submitting them to ISP.

GEMSS can handle both selective (triggered by a periodic scan of StoRM bring-online table or requested through GEMSS command *yamssEnqueueRecall*) and transparent recalls (triggered accessing the files). GEMSS server is able to transform transparent recalls in selective ones.

Figure 1 shows the selective tape-ordered recall system of GEMSS. First, requests are enqueued by a FIFO (First In First Out) method. The *yamssReorderRecall* process builds, for each tape, a list of files to recall sorted according to tape ordering. A recall process (*yamssProcessRecall*) can start for each tape file, according to the GEMSS configuration: for each file system, the maximum number of recall threads to send to ISP server is defined by the parameter *RECALL_RUNNING_THREADS*. Each running recall thread corresponds to a tape drive devoted to read the requested files. In the same way, the *MIGRATE_RUNNING_THREADS* parameter stabilizes the maximum number of running migration threads for each file system. Once the number of running recall threads hits the value of *RECALL_RUNNING_THREADS* parameter, all the other tape files are put in a queue. In case of new requests, *yamssReorderRecall* can add new files to the existing lists in the correct order. The *yamssMonitor* service is the supervisor of the reorder and recall phases. It discovers managed Spectrum Scale file systems on HSM nodes, reads the configuration file for each file system and triggers the needed actions, e.g. starting other processes. It loops continuously in background.

Within a file system, the criterion to assign priority to tapes to recall is given by the *RECALL_MAX_RETENTION* parameter (default value is 1800 seconds). In case pending recall threads waiting time is equal or lower than *RECALL_MAX_RETENTION*, priority is given to those

tapes containing the largest number of files. Instead, if pending recalls threads waiting time is greater than *RECALL_MAX_RETENTION*, then priority is given following a FIFO method.

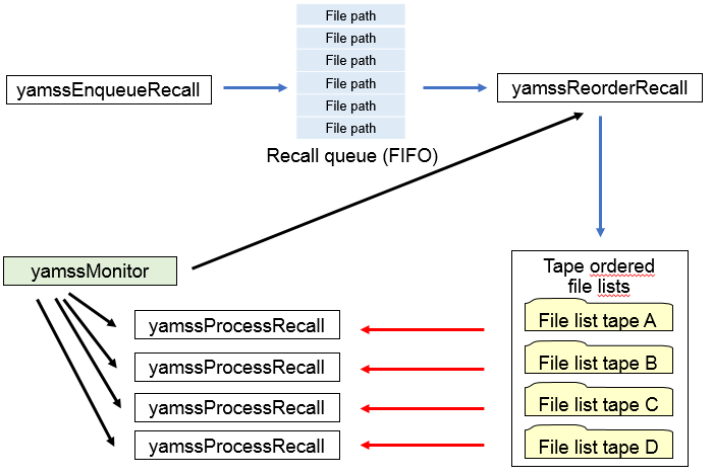

Figure 1. GEMSS recall system

2. Current status in tape drive usage

With the current configuration, the 17 T10000D tape drives dedicated to scientific data are shared among experiments, i. e. any of them can be used by migration or recall threads running for each file system. A maximum number of drives for recalls or migrations, statically defined in GEMSS, can be exploited by each file system. In case of scheduled massive recall or migration activity, these parameters are manually changed by administrators. There is no way to automatically change them.

Sometimes we notice a certain number of free drives that could actually serve pending recall threads, as shown in Figure 2 (plot on the left). In several cases, a subset of free drives could be profitably used to reduce the queue of pending recalls. However, the maximum number of possible running threads is limited by the HSM server throughput capacity. Currently each HSM server is equipped with a single FC8 (Fiber Channel 8 Gbit/sec) connection to the Tape Area Network, so it is capable of handling 800 MB/s simultaneously for inbound and outbound traffic. Given each T10000D tape drive can reach ~200-250 MB/s of throughput, at the moment each HSM server is able to support up to 4 migration and 5 recall processes, considering some observed inefficiencies in recalls due to the not-subsequent placement of files on tape. The HSM connection is planned to be upgraded to FC16 for each server next year.

In case of concurrency in the usage of drives, i. e. when recall or migration threads for one or more file systems cannot become running because of the lack of free drives, there is no way to dynamically change GEMSS parameters to give more priority to file systems that less used the system in the recent past.

Figure 2 (plot on the right) shows duration of recall and migration processes for the overall infrastructure, aggregated by day. The total usage is never greater than 8 days. This means that for the period of the plot, if we do not consider peaks, i. e. intervals of time with a drive usage above average, we could perform migration and recall activity with only 8 drives.

Moreover, in the next years, the usage of tape drives is expected to become more intense due to the growing amount of scientific data and to the trend, already disclosed by the main user communities, to use tapes as near-line (or “slow”) disk, thereby increasing the reading traffic rate.

All of these considerations, together with the need to reduce hardware purchases, moved us to reflect on a drive usage optimization that would reduce their inactivity periods.

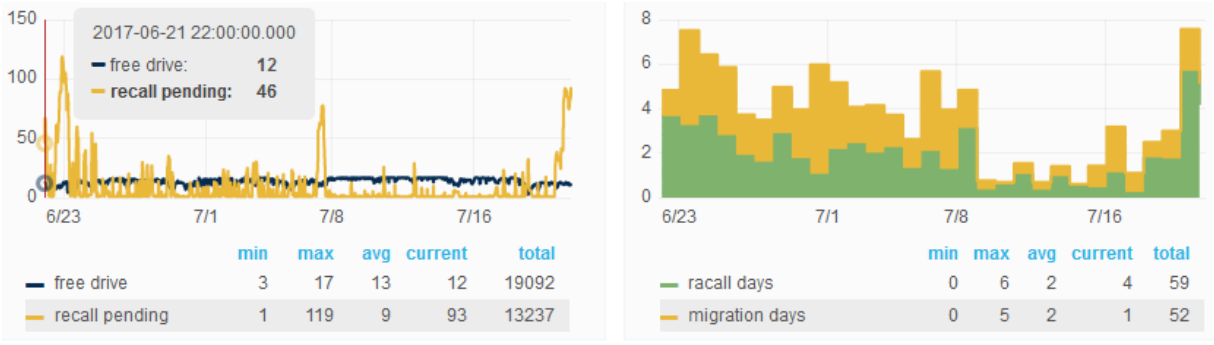

Figure 2: On the left: total number of recall threads pending and free drives. On the right: total duration (in days) of recall and migration processes, stacked plot aggregated by day. June-July 2017.

3. Dynamic sharing of tape drives

Given the weaknesses observed in the static-priority assignment of tape drives, we designed a software solution, hereafter named *Orchestrator*, which allows to dynamically allocate additional drives to file systems, in case free drives are there, and to improve on the management of concurrent recall accesses from different file systems.

Two new GEMSS parameters have been defined: *RECALL_MAX_RUNNING*, representing the maximum number of possible running recalls for each file system, taking into account the FC connection limits of the relevant HSM server; *RECALL_DEFAULT_RUNNING*, representing the value that should have the *RECALL_RUNNING_THREADS* parameter in normal conditions.

The *Orchestrator* uses the library InfluxDB-Python as a client for accessing InfluxDB and read its data. Indeed, monitoring information, essential for the *Orchestrator* to operate, is stored in InfluxDB and updated every five minutes. More specifically, such information includes the number of drives that are currently free or in use (taken from the ISP server), the number of running recall and migration threads, the number of pending recalls and the value of *RECALL_MAX_RUNNING* parameter for each file system (taken from the HSM servers). As a first step of the algorithm, the *Orchestrator* extracts all these relevant quantities from InfluxDB (Figure 3). Like every priority-driven algorithms, the *Orchestrator* performs on-line scheduling, so it makes decision without any knowledge about the kind and amount of workload that will come in the future.

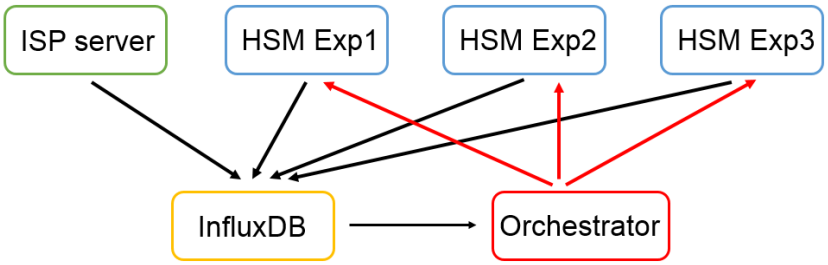

Figure 3. Monitoring information flows from ISP server and HSM servers to InfluxDB (black arrows). The *Orchestrator* reads it and updates GEMSS parameters in the HSM servers (red arrows).

Every five minutes, the *Orchestrator* inspects monitoring information. In case there are free drives and pending requests waiting for drives for any handled file system, it establishes the number of drives that can be assigned to each interested file system, comparing the number of actual running recalls with the value of *RECALL_MAX_RUNNING* parameter. Moreover, in order to exclude situations of ongoing rearrangements of tape drives, it checks whether the number of running threads equals the number of drives in use.

Whenever there is no concurrency among different file systems for the available resources, i.e. the number of available free drives is large enough to satisfy all the requests, the *Orchestrator* mitigates the pending requests. This is performed by means of modifying on the relevant HSM server the GEMSS parameter *RECALL_RUNNING_THREADS* for each interested file system, and by raising it to the value of *RECALL_MAX_RUNNING*. Of course, ideally one would like to lower the number of pending recall and migration threads by filling all the available free drives except for a reasonable reserve (which we set to 2, given the total of 17 drives).

Instead, a more complicated situation can happen in case the pool of free drives is not sufficient to satisfy all the requests. Moreover, it is also possible that pending recalls for a certain file system can not become running due to all the drives being busy with other operations. All these cases of concurrent access to tape drives are managed by the *Orchestrator* by computing and setting a dynamic priority for each file system on the basis of the following formula:

$$FSpriority = FSshare/(\alpha(usage\_time) + \beta(1 + run\_recall))$$

where *FSshare* is a static priority given to each file system, *usage_time* is the total recall time used by the file system in a fixed period of recent past (e.g. last 24 hours), *run_recall* is the number of recall running threads, and finally α and β are adjustable coefficients which allow to differently weight resources usage time in the past (*usage_time*) and current usage time (*run_recall*).

Once the file systems are placed in one common priority queue according to the values of *FSpriority*, the available free drives are assigned going through the sorted list. Then, when all the drives are occupied and new requests become pending, the value of *RECALL_RUNNING_THREADS* can be lowered for a given file system and increased for the file system with pending recalls. In practice, this procedure increases the priority of getting a tape drive for those file systems that did not extensively use resources in the recent past, and who are not currently performing many recalls and migrations.

In any case, when the actual number of running recall threads for a file system is equal or lower than *RECALL_DEFAULT_RUNNING*, the parameter *RECALL_RUNNING_THREADS* is brought again to the default value (*RECALL_DEFAULT_RUNNING*) by the *Orchestrator*.

As noticed in paragraph 2.1, by setting *RECALL_MAX_RETENTION* parameter it would possible to give priority to certain pending recalls. Of course, it would be interesting to consider such procedure for future *Orchestrator* implementations in order to provide a different priority method to dedicated recall threads.

4. Conclusions

CNAF mass-storage infrastructure is handling tens of PB of scientific data. Data movements from disk to tape and vice versa are optimized by means of GEMSS software. In order to overcome the static assignment of a maximum number tape drives both for migration and recall processes, we designed a software solution to dynamically allocate additional drives to file systems and to manage concurrent requests. This solution is expected to optimize the tape drives usage, reducing migration and recall waiting time, that would be an important enhancement for CNAF mass-storage facility in view of the future growth of writing and reading rate. Moreover, the ability to maximize the drive exploitation would help CNAF in lowering hardware purchase, by reducing the need to purchase more tape drives in the future.

References

[1] IBM Spectrum Scale web page: www.ibm.com/systems/storage/spectrum/scale

[2] IBM Spectrum Protect web page: www.ibm.com/systems/storage/spectrum/protect

[3] R. Zappi R et al, *An efficient Grid data access with StoRM*, S.C. Lin and E. Yen (eds.), Data

Driven e-Science: Use Cases and Successful Applications of Distributed Computing Infrastructures (ISGC 2010), Springer Science + Business Media, LLC 2011

[4] Bonacorsi D et al., *The Grid Enabled Mass Storage System (GEMSS): the Storage and Data management system used at the INFN Tier1 at CNAF*, 2012 J. Phys. Conf. Ser. 396 042051 Proceedings of 2012 CHEP conference.