The GridKa tape storage: latest improvements and current production setup

**Haykuhi* Musheghyan[1], *Samuel* Ambroj Pérez[1], *Andreas* Petzold[1], *Doris* Ressmann[1], and *Jan Erik* Sundermann[1]

[1]Karlsruhe Institute of Technology

Abstract.
Tape storage remains the most cost-effective system for safe long-term storage of petabytes of data and reliably accessing it on demand. It has long been widely used by Tier-1 centers in WLCG. GridKa uses tape storage systems for LHC and non-LHC HEP experiments. The performance requirements on the tape storage systems are increasing every year, creating an increasing number of challenges in providing a scalable and reliable system. Therefore, providing high-performance, scalable and reliable tape storage systems is a top priority for Tier-1 centers in WLCG.
At GridKa, various performance tests were recently done to investigate the existence of bottlenecks in the tape storage setup. As a result, several bottlenecks were identified and resolved, leading to a significant improvement in the overall tape storage performance. These results were achieved in a test environment and introduction of these achievements in to the production environment required a great effort, among many other things, a new software had to be developed to interact with the tape management software.
This contribution provides detailed information on the latest improvements and changes on the GridKa tape storage setup.

1 Introduction

Most Tier-1 centers in the Worldwide LHC Computing Grid[1], including GridKa (German Tier-1 center in Karlsruhe) support both disk and tape storage systems. Tape storage systems remain the most cost-efficient solution to store large volumes of infrequently used data. The demand for its usage grows from year to year and new creative and easily adjustable approaches and solutions are required to satisfy the growing requirements. Expansion of storage systems complicates their maintenance and support.

From mid-2018 to mid-2019, several performance tests were done on the GridKa tape storage system to identify bottlenecks in the setup and fix them if possible. These tests produced the expected results and several bottlenecks were discovered and eliminated. Eliminating the discovered bottlenecks significantly improved the overall performance of the tape storage system. All this work was done in a test environment and details about the results of various performance tests can be found in the proceedings of CHEP2019[2].

Implementing these results in production required special efforts, such as developing new software, modifying existing configurations, replacing old hardware and upgrading network connections.

© The Authors, published by EDP Sciences. This is an open access article distributed under the terms of the Creative Commons Attribution License 4.0 (http://creativecommons.org/licenses/by/4.0/).

The latest changes are described in the following sections of this article.

2 GridKa storage overview

For over 10 years, GridKa has been using IBM Spectrum Protect (IBM SP)[3], formerly known as Tivoli Storage Manager (TSM) as a tape storage system and TSS-client[4] as a TSM-based tape client. The TSS-client is developed by the GridKa team. GridKa uses one Oracle SL8500 as a tape storage library.

For many years, GridKa has been using dCache[5] as a disk storage system. Compared to other WLCG sites, GridKa has a slightly different dCache setup. In GridKa, dCache pools are directories in a large IBM Spectrum Scale[6] file system, and pool sizes are logical settings. With regard to dCache tape pools, there is only one primary and one fallback stage and write pool. The dCache stage pool is used to stage files from tape, and the write pool is used to flush files to tape.

In the old setup, the interaction between the GridKa dCache instance and IBM SP has been accomplished via additional scripts as shown in Figure 1 (up). The setup had only one host and all components like dCache pools, scripts, and the TSS-client were running on that host consuming high memory. This workflow worked quite well to handle up to 2.000 simultaneous requests. but beyond of that it faced/ran into out-of-memory problems and as a result ended up with a crash of the dCache tape pool.

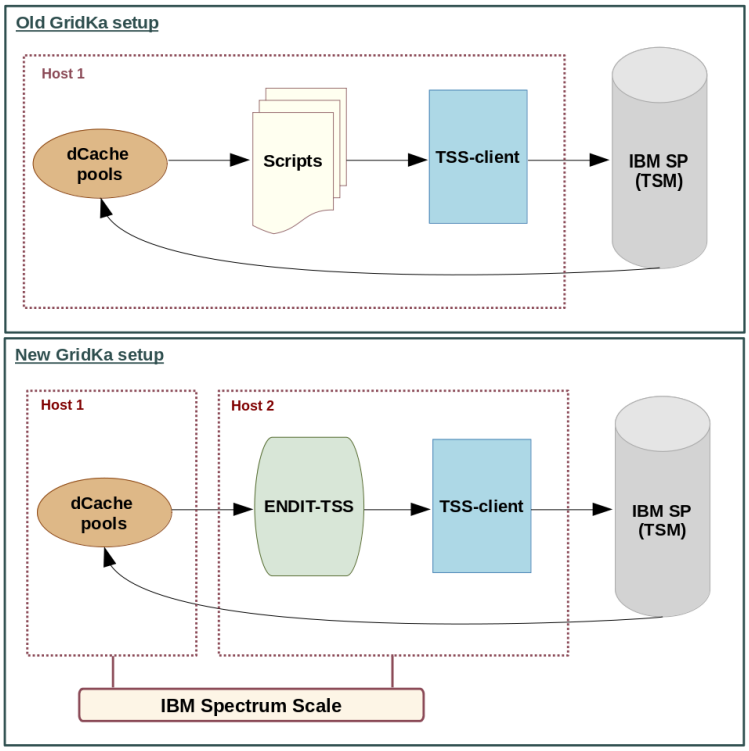

Figure 1: Gridka old(up) and new (bottom) setups: Interaction between dCache tape pools and IBM SP

Having only one dCache stage and one write pool is needed to optimize tape throughput. The TSS-client sorts files by tape cartridges to avoid long seek times. It works on a particular host and builds tape queues for requests arriving to a particular dCache tape pool configured on that host. If, for example, 2 primary dCache stage pools are configured on 2 different hosts, then 2 TSS-clients must be configured on these hosts, one for each host. Then, stage requests arrive, and tape queues are built independently of the hosts. If stage requests try to stage files located on the same tape, it creates contention between the tape drives. To avoid this situation, only one primary dCache tape pool is configured on GridKa for each activity (stage or flush). Consequently, it is very important to have as many simultaneous requests as possible.

During various performance tests, it was found that the number of simultaneous requests could be increased from 2.000 to 30.000. Within the workflow of the old GridKa setup, described in Figure 1 (up) this would not have been possible due to the limitations described above. Respectively, the workflow needed to be modified, as shown in Figure 1 (bottom).

On the dCache side, GridKa started using a nearline storage plugin[7] called dCache Endit-Provider plugin[8]. The dCache Endit-Provider plugin gives the possibility to process tens or hundreds of thousands of requests simultaneously. The way it is designed allows to handle flush or stage requests simultaneously. These requests arrive and stay in the appropriate dCache Endit-Provider directory and wait until they are processed by the tape storage system. In the old setup, the dCache hsm script[9] was used instead of this plugin.

The additional scripts shown in Figure 1 (up) have been substituted with the dCache Endit-Provider plugin and new software called Endit-TSS. See Sections 3 and 4 for more details.

On the TSS-client side, the maximum number of simultaneous requests was increased from 2.000 to 30.000, as was done in the test environment.

In the new setup, the high memory consumption issue is solved by distributing these components across two hosts, connected to each other via IBM Spectrum Scale. One host runs dCache pools with the Endit-Provider plugin, and the other host runs Endit-TSS software and TSS-client.

The new GridKa setup has been in production since January 2020 for ATLAS and Belle2 and from the beginning of 2021 also for the CMS and LHCb experiments.

3 dCache nearline storage plugins

3.1 dCache nearline storage

The dCache storage system provides a number of different plugins for different purposes, and one of them is a tape-like storage plugin or nearline storage plugin. The dCache pools may copy files to nearline storage to write (“flush”) them to tape and read these (“stage”) files whenever the need arises.

A nearline storage driver supports three types of requests: flush, stage, and remove. The request object describes the file to be flushed, staged, or removed, and acts as a callback through which the driver reports a status change, completion or failure of the request. Each request object has a unique identifier (ID), and dCache may cancel the nearline request at any time using the corresponding ID.

3.2 dCache Endit-Provider plugin

The dCache Endit-Provider plugin interacts with the nearline storage driver. It is organized through different directories such as “in”, “out”, “request” and “trash” under the pool’s data

directory. Depending on the particular action “stage”, “flush” or “remove”, the corresponding directory is used. The "request" directory is used for storing metadata files. A metadata file is a file containing a JSON object. It has fields such as file size, time and other necessary information that is used for staging or flushing a file from or to tape respectively.

The plugin was developed by NDGF-T1[10]. The original source code of the plugin has been slightly modified by the GridKa team to adapt to the GridKa use case. In particular, a few more fields have been added to the JSON object of the metadata file, such as the storage class, file path, and action. These fields are necessary to build the correct command for the TSS client. In the flushing part, the creation of a metadata file was not implemented, so it was added.

The workflow of the modified version of the plugin is shown in Figure 2.

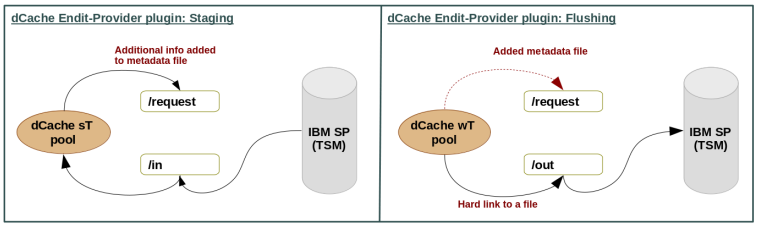

Figure 2: The workflow of dCache Endit-Provider plugin (slightly modified by GridKa)

Staging a file from tape: The left side of Figure 2 shows the workflow of staging a file from tape. If a file needs to be staged from tape, the corresponding metadata file will be written to the “request” directory, and dCache waits until the file is staged into the “in” directory. After the file has been staged with the correct name and file size, dCache Endit-Provider plugin will move the file from the “in” directory to the data pool and remove the metadata file from the “request” directory.

Flushing a file to tape: The concept of hard links is used here because it is convenient. Hard links remain linked even if the original files are moved across the same file system. In case the original file is removed, the link will still show the contents of the file, preventing the possibility of data loss.

The right side of Figure 2 shows the workflow of flushing a file to tape. If a file needs to be flushed to tape, the corresponding metadata file will be written to the “request” directory, the file’s hard link to the "out" directory. If a file is somehow gone/deleted from the “out” directory, the plugin assumes that the file was successfully flushed to tape, although the file may have been manually deleted by accident. The plugin does not check or verify, if the file was actually flushed to tape or not. If a hard link is removed from the "out" directory, the corresponding metadata file will also be removed from the "request" directory. The original file remains on the dCache pool, regardless of whether the file flushing was successful or not.

In the case of GridKa, the dCache Endit-Provider plugin interacts with the Endit-TSS software.

4 Endit-TSS software

Endit-TSS is an intermediate software that works in conjunction with the dCache ENDIT-Provider plugin and the TSS-client. It is a new software developed by the GridKa team in 2019. Endit-TSS is written in Java.

Staging a file from tape:

Figure 3 displays, the entire file staging workflow for the GridKa use case. If a file needs to be staged from tape, a corresponding metadata file is written to the "request" directory. Endit-TSS simply monitors the "request" directory. As soon as the metadata file appears in the "request" directory, Endit-TSS starts processing the metadata file. It extracts its contents, constructs the appropriate TSS stage command, and creates a TSS stage request. The TSS-client receives the stage request, sorts it by the tape cartridge, builds a queue and starts staging the file from tape. The file is staged directly to the "in" directory. If the file is staged from tape successfully, the dCache ENDIT-Provider plugin will take care of the rest as described in Section3.2. If the file is staged unsuccessfully from tape, for example, if the tape drive is defective, then the stage request will remain active until it is fixed so that the file can be processed.

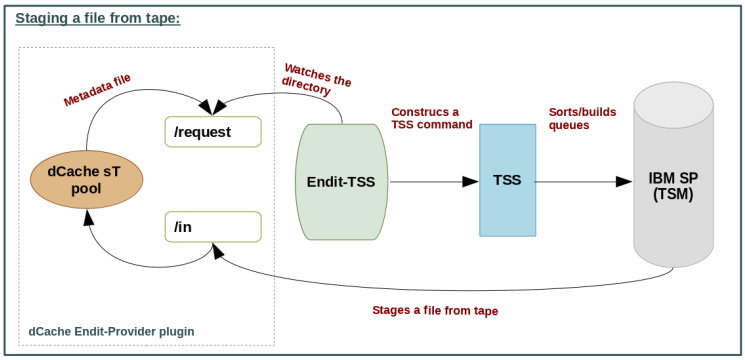

Figure 3: Staging a file from tape using the dCache Endit-Provider plugin and Endit-TSS software

Flushing a file to tape:

Figure 4 displays, the entire file flushing workflow for the GridKa use case. If a file needs to be flushed to tape, a corresponding metadata file is written to the "request" directory and the file hard link to the "out" directory. Endit-TSS monitors the "request" directory. As soon as a metadata file appears in the "request" directory, Endit-TSS starts processing it. It extracts its contents, constructs the appropriate TSS flush command, and creates a TSS flush request. Then, the TSS-client takes a flush request, sorts and builds a queue and starts flushing the file to tape. If a file is successfully flushed to tape, Endit-TSS removes the corresponding hard link of the file from the "out" directory. The rest is taken care of by the plugin. If the file flushing to tape was unsuccessful, for example in case of a problem with tape cartridges or tape drives, the flush request will remain active until the request is processed.

Having been in production for some time now, Endit-TSS has proven its effectiveness. The results are presented in Section5.

5 Latest results and conclusions

In the new setup, all identified bottlenecks have been eliminated. The combination of all of the above changes as well as the elimination of all identified bottlenecks guaranteed a significant improvement in the overall tape storage performance also in production mode at GridKa.

These improvements are visible in Figure 5. It displays the tape storage stage or recall throughput within the old and new setups in production mode, respectively. In the old setup,

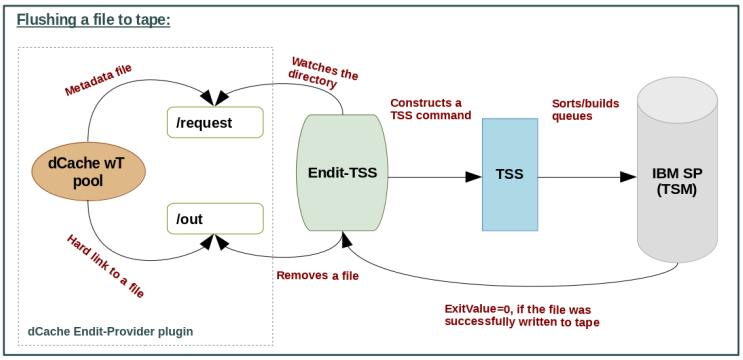

Figure 4: Flushing a file to tape using the dCache Endit-Provider plugin and Endit-TSS software

the average tape staging/recalling throughput was ≈350MB/s for 8 tape drives, which corresponds to ≈40MB/s per drive, while the new setup allows to get ≈1100MB/s for 8 drives or ≈135MB/s per drive. This is a very good achievement. The new setup provides more than factor of 3 improvement in the tape storage stage/recall performance.

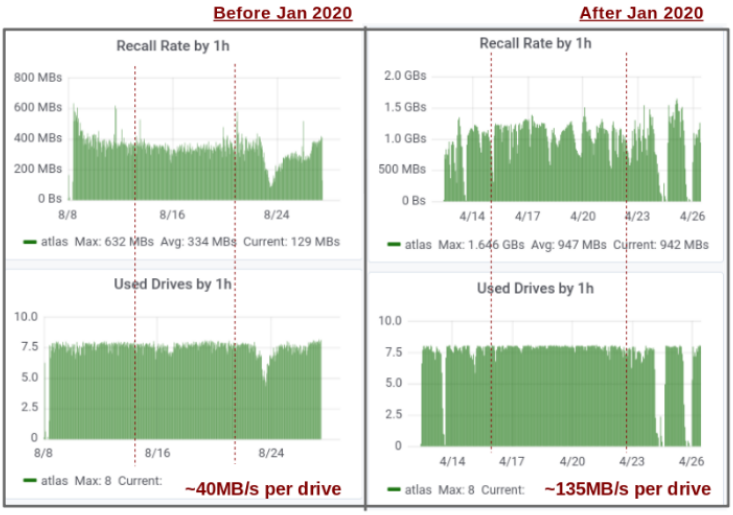

Figure 5: Tape storage performance comparison results for old and new GridKa setups

The new setup guarantees more efficient and optimized usage of the GridKa tape storage system. These results were expected.

Currently, the GridKa team is monitoring new results that appear as a result of recent changes in the GridKa setup, and collecting statistics from various experiments for further possible improvements.

No further improvements are planned for IBM SP in the future as GridKa plans to migrate to a different tape storage system. GridKa is preparing to migrate from IBM SP to High

Performance Storage System (HPSS)[11], and intensive work is currently underway in this direction. Once this is done, all of the above mentioned changes will be adapted to the needs of HPSS.

References

[1] Wordwide LHC Computing Grid, https://wlcg.web.cern.ch/

[2] Musheghyan, Haykuhi, et al. "The GridKa Tape Storage: various performance test results and current improvements." EPJ Web of Conferences. Vol. 245. EDP Sciences, 2020.

[3] IBM Spectrum Protect (IBM SP), https://www.ibm.com/support/knowledgecenter/de/SSEQVQ/landing/welcome_sseqvq.html

[4] J. van Wezel, D. Ressmann, "TSM as tape storage backend for disk pool managers", HEPiX Spring Meeting, Roma, I, April 3-7, (2006)

[5] dCache storage system, https://www.dcache.org/

[6] IBM Spectrum Scale, https://www.ibm.com/products/spectrum-scale

[7] dCache nearline storage plugins, https://dcache.org/old/manuals/Book-6.1/cookbook-writing-hsm-plugins.shtml

[8] dCache Endit-Provider plugin, https://github.com/neicnordic/dcache-endit-provider

[9] The dCache Tertiary Storage System Interface, https://dcache.org/old/manuals/Book-5.0/config-hsm.shtml

[10] NDGF-T1, https://neic.no/nt1/

[11] High Performance Storage System (HPSS), http://hpss-collaboration.org