Enhancing data integrity and resilience: extending the CERN backup system with a tape-based backend

Gianmaria Del Monte[1,*], *Hugo* Labrador Gonzalez[1,**], *Fons* Rademakers[1,***], and *Roberto* Valverde[1,****]

[1]CERN - European Organization for Nuclear Research, 1211 Geneva, CH

Abstract. The CERN IT Department is responsible for ensuring the integrity and security of data stored in the IT Storage Services. General storage backends such as EOS, CERNBox and CephFS are used to store data for a wide range of use cases for all stakeholders at CERN, including experiment project spaces and user home directories.
In recent years a backup system, cback, was developed based on the open source backup program restic. cback is currently used to backup about 2.5 billion files and 18PB stored on disks in the CERN Computing Center.
To significantly increase the reliability and security of the backups and reduce the storage costs, by limiting the amount of data on disk, we have added a tape storage backend to cback.
With this addition cback can reliably be extended to new use cases, like backing up any local mountable file system, such as EOS, CephFS, NFS or DFS.
In this paper we will describe the architecture and implementation of cback with the new tape storage backend and a number of developments planned for the near future.

1 Introduction

At CERN, the European Organization for Nuclear Research, where massive volumes of critical data are generated, the storage group assumes the responsibility of ensuring the secure and dependable storage of the physics data. This responsibility includes managing and providing services for different storage solutions, including CERNBox [1], Ceph, EOS [2], DFS, AFS, CVMFS and CTA (CERN Tape Archive) [3], each tailored to meet different specific demands of the CERN community. In this context, the cback system has been developed, to provide a reliable backup service for some of these critical services. cback is a backup orchestrator that internally uses *restic* [4], an advanced, open source, backup program supporting snapshots, deduplication, encryption, compression, snapshot pruning and many different storage backends. Currently deployed for backing up data to S3, served by Ceph, stored in another geographical location at CERN, cback can backup any local mountable file system, like EOS, CephFS, NFS, among others.

*e-mail: gianmaria.del.monte@cern.ch
**e-mail: hugo.gonzalez.labrador@cern.ch
***e-mail: fons.rademakers@cern.ch
****e-mail: roberto.valverde.cameselle@cern.ch

While the current backup system stores efficiently data on hard disks, an important missing component was a connection to a *cold storage* facility. As CERN continues to generate vast quantities of scientific data, the demand for storage capacity is increasing. Therefore, the cost of maintaining a large-scale, high-performance disk storage infrastructure for archiving infrequently accessed data becomes prohibitively high over time.

Furthermore, data resilience in case of catastrophic events, like natural disasters, human-made incidents or ransomware attacks, is of paramount concern for CERN. Hard disks are susceptible to various forms of corruptions, making them less reliable as a disaster recovery solution [5, 6].

To comprehensively address these challenges, the introduction of a cold storage layer becomes imperative. Cold storage offers several key advantages in mitigating the limitations of a disk only system. Notably, as tape is still quite a bit cheaper than disk, it provides an economical solution for storing large volumes of data that are accessed infrequently, thus significantly reducing operational costs. Moreover, the inherent durability of tape technology makes it an ideal choice for safeguarding data against corruption and ensuring a robust disaster recovery mechanism.

At CERN, cold storage is provided by a large robotized tape library facility controlled by the CERN developed CTA service. In the following sections, we describe the architectural details of cback and its integration with CTA.

2 The cback system architecture

2.1 restic – the backup engine

restic [4] is a powerful and open-source backup software designed to provide efficient, secure and flexible data protection. restic features a command-line interface (CLI) that offers users full control over backup and restore operations. It combines features like incremental backups, efficient data de-duplication, robust encryption. Its main characteristics lies in its approach to data de-duplication thanks to *content-defining chunking* [7]. Unlike conventional backup systems that rely on fixed block sizes, restic dynamically divides data into variable-sized chunks based on its actual content. These chunks are accompanied by cryptographic fingerprints, enabling restic to identify and store identical data chunks only once, reducing redundancy and optimizing storage efficiency.

restic employs a snapshot-based approach for backup management. It organizes backups into repositories, with each repository containing a series of snapshots representing different points in time. The layout of a restic repository is shown in the figure 1. In particular, notice the `data` directory, housing packs containing file content in the form of chunks, referred as *data pack*, and `tree` objects, referenced by the `snapshots`, containing the structure of the backed up resources.

A retention policy allows users to define how long snapshots should be retained, ensuring older snapshots are automatically pruned based on their defined criteria.

2.2 cback – the orchestrator

cback [8] is a highly scalable system developed at CERN designed to streamline the entire data backup and restore lifecycle, encompassing tasks such as backup, restore, verification and retention policy management. At its core cback leverages restic for managing the backup and restore processes. The system operates through stateless agents (see figure 2), each dedicated to specific functions: backup, restore, prune, verify and sync. Coordination among these agents is orchestrated via a central database containing job definitions.

```
/tmp/restic-repo
├── config
├── data
│   ├── 21
│   │   └── 2159dd48f8a24f33c307b750592773f8b71ff8d11452132a7b2e2a6a01611be1
│   ├── 32
│   │   └── 32ea976bc30771cebad8285cd99120ac8786f9ffd42141d452458089985043a5
│   ├── 59
│   │   └── 59fe4bcde59bd6222eba87795e35a90d82cd2f138a27b6835032b7b58173a426
│   ├── 73
│   │   └── 73d04e6125cf3c28a299cc2f3cca3b78ceac396e4fcf9575e34536b26782413c
│   [...]
├── index
│   ├── c38f5fb68307c6a3e3aa945d556e325dc38f5fb68307c6a3e3aa945d556e325d
│   └── ca171b1b7394d90d330b265d90f506f9984043b342525f019788f97e745c71fd
├── keys
│   └── b02de829beeb3c01a63e6b25cbd421a98fef144f03b9a02e46eff9e2ca3f0bd7
├── locks
├── snapshots
│   └── 22a5af1bdc6e616f8a29579458c49627e01b32210d09adb288d1ecda7c5711ec
└── tmp
```

Figure 1. The restic repository layout.

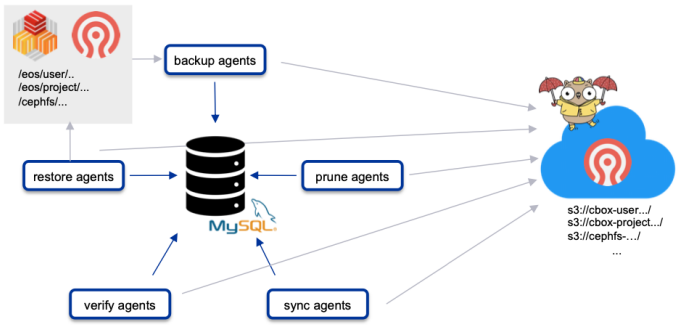

Figure 2. The CBACK Architecture.

Unlike traditional backup systems, cback operates as an always-running backup solution. Its agents continuously look for new tasks within configured intervals, ensuring a distributed backup workload over time.

The main component of cback is the *backup agent*, which is responsible of executing the backup jobs. The restore jobs are handled by the *restore agent*. The *prune agent* performs pruning and purging based on a configured retention policy. The *verify agent* is responsible for performing the backup verification jobs, consisting on integrity checks on the data stored in the restic repository. Additionally, the *sync agent* is responsible of ensuring synchronization between two or more restic repositories, particularly when they are stored in distinct geographical locations.

Thanks to restic, cback supports various backends and can seamlessly backup data from any local mountable storage. At CERN, for example, it efficiently handles data from diverse sources such as CERNBox, EOS, CephFS and CVMFS, backing up this data to S3, stored in Ceph.

In cback, jobs are organized into *groups*. Groups serve as a powerful mechanism for accommodating the inherent diversity among various storage solutions, each with its unique

characteristics and retention policies dictated by service data retention requirements. Each group can be uniquely configured, enabling distinct sets of S3 credentials, restic secrets to be applied. Furthermore, scheduling flexibility withing each group ensures backups are tailored to individual needs. This versatility empowers a single cback service, with various agents in a cluster, to effectively backup different storage services, each with its distinct characteristics and retention policies.

3 Tape storage support for cback

3.1 Tape advantages

Although hard disk-based backups have been instrumental in safeguarding essential data, they are insufficient in addressing the diverse and evolving needs of a large scale data intensive organization like CERN. While hard disks excel in terms of speed and accessibility, they may not offer the most cost-effective and resilient solution for the retention of extensive data. Moreover, they can be vulnerable to various risks, such as hardware failure, data corruption, or unforeseen calamities, like, human errors or natural disasters, that could compromise data integrity. Tape-based storage solutions, such as CERN's CTA service, provide a reliable and cost-effective means of preserving large quantities of data over extended periods of time. Tapes have been proven to be resilient to data corruption and physical damage [9–11], positioning them as the preferred choice for disaster recovery scenarios and long-term archival. Furthermore, tapes cost significantly less per terabyte compared to traditional hard disk solutions.

3.2 Challenges and limitations

This section outlines two straightforward solutions for integrating cold storage into the cback system. However, it is important to recognize certain limitations of each approach. In the first approach, restic packs are directly archived onto tape with the default pack size adjusted to 16MB. While this method may appear simpler on the surface, it introduces complexities related to how files are positioned on tape. As restic packs are added incrementally over time, this process could result in pack fragmentation within the tape storage system. Packs may become distributed across multiple tape mounts, leading to an increased seek time during data retrieval, and impacting the efficiency and speed of the restore process.

Alternatively, the process can be broken down into distinct stages: 1) *snapshot extraction*, where according to a predefined policy, selected snapshots are extracted from the original restic repository into a new repository. This newly created repository serves as a container for preparing the data for archival. 2) *repository compression*, where the entire repository folder is compressed in a single archive. 3) *segmentation of the compressed archive*, where the repository is divided in smaller segments based on a predefined maximum file size threshold. 4) *archival to tape*, where the segments are send to the tape system. cback concludes the process by marking the snapshots as stored in cold storage and purging the corresponding data from the hard disks. However, some risks are associated with this approach. For instance, in the event of segment corruption, the loss will extend to the entire repository and all the data contained within it. Additionally, archives may necessitate periodic updates and maintenance to ensure continued compatibility with restic. Furthermore, since each snapshot is treated as a distinct entity, the storage gains achieved through repository-wide de-duplication are diminished. This could result in increased storage requirements over time on the tape system.

3.3 An efficient archival method

As discussed in the previous section, it becomes clear that 1) the native format of the restic repositories is not suitable for a tape system, and 2) there should be some mechanism of de-duplication to optimize storage utilization.

To address these challenges, we introduce an archival method within the cback framework. This method is controlled by a specialized component known as the *archiver agent*. It works in accordance with the directives of the *archive policy engine* (see the section 4 for detailed information), configured for each group. The policy engine selects the snapshots to be archived. A snapshot can be selected for a *full* or *partial* archive. A full archive will contain all the resources in the snapshot. A partial archive focuses only on the differences between the selected snapshot and its preceding counterpart. It is important to notice that in partial archives, resources can be added, deleted or updated (metadata and/or data). All other operations, like a move or copy on the original source, can be converted into a series of these fundamental operations. Notice also that a full archive can be seen as a partial with a (virtual) empty previous snapshot and only with added resources. The archiver agent considers the resources contained in the selected snapshots, which are easily accessible thanks to the `restic mount` command. In case of a partial snapshot, the resources to consider are obtained using the `restic diff` command (see figure 3).

```
> restic diff dcef7dbf 45b2d489
repository ad1f458b opened (version 1)
comparing snapshot dcef7dbf to 45b2d489:

+    /eos/home-g/gdelmont/added
-    /eos/home-g/gdelmont/removed
M    /eos/home-g/gdelmont/updated

Files:           3 new,     0 removed,     0 changed
Dirs:            0 new,     0 removed
Others:          0 new,     0 removed
Data Blobs:     22 new,     0 removed
Tree Blobs:      9 new,     9 removed
  Added:   15.815 MiB
  Removed: 17.755 KiB
```

Figure 3. An example of the `restic diff` command.

The archiver agent assembles a set of zip archives containing each some of the resources from the snapshots. Deleted files are appropriately flagged within the archive. These archives are created in order to adhere to specific size criteria, exceeding a minimum threshold while remaining below a defined maximum threshold. Once the archive is ready, it is dispatched to the tape system. Notably, due to potential minimal differences between snapshots, an archive can contain resources from different restic repositories, belonging to the same group. Each resource within the archive retains its full path. Archives stored on tape are named based on a specific convention. For each backup and snapshot included in the archive, a string is constructed by concatenating `<backup-name>@<snapshot-from-timestamp>!<snapshot-to-timestamp>` tokens, separated by underscores. These archive locations are then recorded within the cback database, accompanied by their corresponding snapshot and backup identifiers.

We decided on storing on tape zip archives, a very well described and simple archive format including a simple file index, instead of the more complex data chunked restic repositories. Zip archives allow for quick single file or directory recovery without having to worry about an appropriate version of restic being available in the medium or far future.

This method solves the problem that we found in the previous section: 1) unmodified data are not stored again, optimizing storage efficiency on tapes, 2) archives are self-contained, meaning that in case of archive corruption, only the files within the affected archive are impacted, and 3) the fragmentation of resources on tape is limited.

This approach introduces the potential for asynchronous restore, contingent on the availability of archives for processing and subsequent discarding once retrieval is complete. Another advantage emerges in case of disaster recovery: if the cback database is lost, it can be easily reconstructed by examining the tape system namespace.

3.4 Separating metadata from a restic repository

Currently restic employs a unified repository structure containing both data and metadata, stored on a fast and accessible storage medium, typically hard disks or solid-state disks. In our work, we have extended restic's capabilities to support the creation of repositories containing only metadata, referred as *metadata-only repositories*. Metadata includes file and directory attributes, the index of the chunks, snapshot attributes and everything that is not data pack file. This separation of metadata from data allows for the creation of a distinct repository dedicated solely to metadata, while the actual data content remains in a conventional restic repository, referred to as *full-repository*. This ensures that vital information such as chunk references, snapshot organization, file browsing and lookup, remains accessible.

To facilitate this metadata-only repository setup, the restic backup command introduces an option named `--repo-hot`, for the metadata-only repository. Users can specify this option along with the location of the full-repository. During the backup process, when this option is specified, restic refrains from reading any data from the full repository while preserving duplication capabilities, thanks to the metadata-only repository.

The initial implementation resides within a fork of restic, inspired by a work already started by the restic community [12]. However, we are exploring the possibility to extrapolate this logic thanks to the restic REST APIs [13].

4 Archive Policy Engine

The decision when to move data from disk storage, to tape storage is taken by the *archive policy engine*. The archive policy engine determines at what rate full and incremental archives will be sent to the tape system. It also controls the data retention policy of the archives stored on the tape system. An example of such a policy is: monthly full and weekly incremental backups with a retention period of two years. Depending on the number and size of the archives these rates can be tuned to not overload the tape system. In the future, more complex rules based on percentages of changed files, snapshot sizes and other criteria can be implemented.

5 Conclusions

In recent developments, we have successfully integrated a cold storage backend into cback, leveraging the CERN-developed CTA tape system. To efficiently manage the transition of backups to tape storage, we've introduced a dedicated policy engine that governs the timing

and frequency of these migrations. This integration elevates cback to a robust and comprehensive backup solution, characterized by its virtually limitless storage capacity. Notably, the tape backend enhances data security by fortifying defenses against ransomware attacks and providing resilience in the face of natural disasters.

Future work for cback includes enhancing user accessibility by introducing a graphical user interface (GUI) (see figure 4) to enable end-users to initiate file restores and perform backups directly within the cback system. This user-friendly interface will expand the usability of cback beyond operators and streamline data management for all end users.

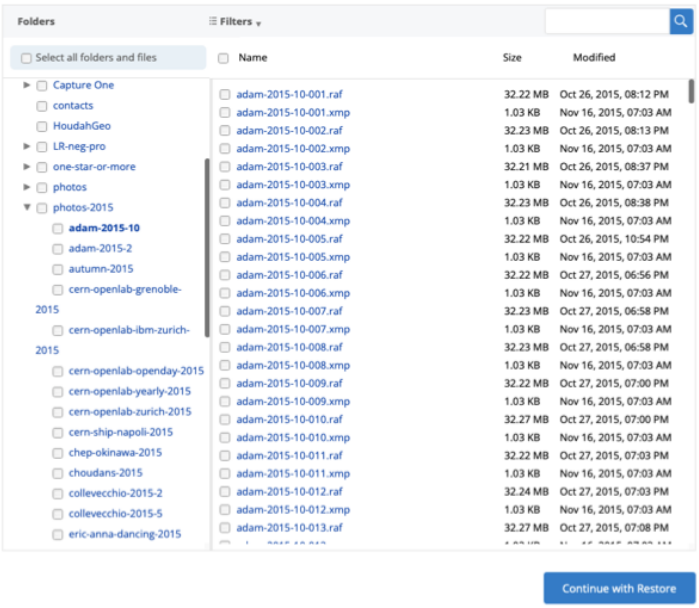

Figure 4. A cback GUI prototype.

Additionally, further improvements to cback will address the current data duplication challenge associated with snapshots stored on disks and tapes. To optimize data storage efficiency, we will explore a hybrid approach where partial snapshots remain on hard disks, while full snapshots are migrated to tape storage. This approach will minimize redundancy.

References

[1] H.G. Labrador, G. Alexandropoulos, E. Bocchi, D. Castro, B. Chan, C. Contescu, M. Lamanna, G.L. Presti, L. Mascetti, J. Moscicki et al., *CERNBox: the CERN cloud storage hub*, in *EPJ Web of Conferences* (EDP Sciences, 2019), Vol. 214, p. 04038

[2] A.J. Peters, E.A. Sindrilaru, G. Adde, *EOS as the present and future solution for data storage at CERN*, in *Journal of Physics: Conference Series* (IOP Publishing, 2015), Vol. 664, p. 042042

[3] E. Cano, V. Bahyl, C. Caffy, G. Cancio, M. Davis, O. Keeble, V. Kotlyar, J. Leduc, S. Murray, *CERN Tape Archive: a distributed, reliable and scalable scheduling system*, in *EPJ Web of Conferences* (EDP Sciences, 2021), Vol. 251, p. 02037

[4] *Backups done right!*, https://restic.net/

[5] J. Elerath, *Hard-disk drives: The good, the bad, and the ugly* (ACM New York, NY, USA, 2009), Vol. 52, pp. 38–45

[6] H. Sandoh, N. Kaio, H. Kawai, Reliability Engineering & System Safety **37**, 29 (1992)

[7] *Introducing content defined chunking (cdc)*, https://restic.net/blog/2015-09-12/restic-foundation1-cdc

[8] R.V. Cameselle, H.G. Labrador, *Addressing a billion-entries multi-petabyte distributed file system backup problem with cback: from files to objects*, in *EPJ Web of Conferences* (EDP Sciences, 2021), Vol. 251, p. 02071

[9] D. Beech, Best Practices for backup and long-term data (2009)

[10] C.Y. Chang, *A survey of data protection technologies*, in *2005 IEEE International Conference on Electro Information Technology* (IEEE, 2005), pp. 6–pp

[11] Y. Okazaki, K. Hara, T. Kawashima, A. Sato, T. Hirano, IEEE transactions on magnetics **28**, 2365 (1992)

[12] *Add possibility to specify extra repo for "hot" data*, https://github.com/restic/restic/pull/3235

[13] *Rest backend*, https://restic.readthedocs.io/en/stable/100_references.html#rest-backend