The Repack Challenge

Daniele Francesco Kruse
IT/DSS group, CERN, Geneva, Switzerland
E-mail: daniele.francesco.kruse@cern.ch

Abstract. Physics data stored in CERN tapes is quickly reaching the 100 PB milestone. Tape is an ever-changing technology that is still following Moore’s law in terms of capacity. This means we can store every year more and more data in the same amount of tapes. However this doesn’t come for free: the first obvious cost is the new higher capacity media. The second less known cost is related to moving the data from the old tapes to the new ones. This activity is what we call repack. Repack is vital for any large tape user: without it, one would have to buy more tape libraries and more floor space and, eventually, data on old non supported tapes would become unreadable and be lost forever. In this paper we describe the challenge of repacking 115 PB before LHC data taking starts in the beginning of 2015. This process will have to run concurrently with the existing experiment tape activities, and therefore needs to be as transparent as possible for users. Making sure that this works out seamlessly implies careful planning of the resources and the various policies for sharing them fairly and conveniently. To tackle this problem we need to fully exploit the speed and throughput of our modern tape drives. This involves proper dimensioning and configuration of the disk arrays and all the links between them and the tape servers, i.e the machines responsible for managing the tape drives. It is also equally important to provide tools to improve the efficiency with which we use our tape libraries. The new repack setup we deployed has on average increased tape drive throughput by 80%, allowing them to perform closer to their design specifications. This improvement in turn means a 48% decrease in the number of drives needed to achieve the required throughput to complete the full repack on time.

1. Repack: a vital activity
What does it mean to repack a tape? Simply put: to move data from that one source tape to one or more destination tapes. At CERN currently, more than 90 PB of physics data is stored on more than 50’000 tapes in 5 libraries. In the following list we describe the reasons why, for large tape users like CERN, repack is of vital importance:

- **Data fragmentation.** When writing to a tape, new data can only be appended to already written data, so when we remove all references to a file on tape, the space allocated to it cannot be used for anything else. Repack addresses tape fragmentation by moving all active files to other tapes, while the source tape is reclaimed and ready to be rewritten from the beginning.

- **Tape media failures.** Enterprise tape is typically regarded as a very reliable data storage media, several orders of magnitude more than disk[1, 2]. However, tapes do get damaged from time to time[3]. Many of these failures do not cause actual data loss, but they are very good predictors of the possibility of more serious problems to come. That is why it is important to repack and dispose of a tape as soon as as a read failure is detected.
• **Tape reallocation.** At CERN tapes are used both for its physics archive and its backup service. Periodically we repack some of the emptiest tapes of the former and give them away to the latter when it is running low on free tapes. Repack thus optimizes the utilization of tapes across internal applications.

• **Tape media evolution.** Data residing on enterprise tapes has an advertised longevity of around 30 years. However, as new higher capacity media and faster tape drive technology becomes available, organizations need to migrate existing data onto new tapes in order to keep the ever growing data contained within the smallest amount of tape slots possible. This allows for potentially huge savings by reducing or avoiding altogether the need to purchase new tape libraries or acquire new floor space.

2. **Repack usage comparison**
At CERN we typically defragment 250 to 700 tapes per year. Problematic tapes that require moving data away from them, amount to 50 to 100 tapes per year. The backup service is instead fed with around 1000 tapes per year. So the first three activities combined require the repack of 1300 to 1800 tapes per year. As far as tape media evolution is concerned, the repack effort is a lot larger: practically every single tape of the archive needs to be repacked onto new media. This, in CERN’s specific case, means moving data out of 50,000 tapes.

3. **The old repack infrastructure and performance**
The old repack disk cache consisted of 17 disk servers, each having 24 disks organized in 12 RAID 1s and 1 Gb ethernet connectivity. Back in 2009 during the last large repack exercise, the CASTOR repack tool was still at an early stage and the average recall speed was 56 MB/s, while drives and network allowed more than the double[4]. In 2013 the tests performed showed a peak recall performance of 1.8 GB/s, with 20 drives recalling in parallel, averaging 90 MB/s per drive. The migration speed instead averaged 140 MB/s per drive. This configuration was clearly inadequate as it would have required around 60% of all the available drives just for the repack process to perform at the required throughput of 3.8 GB/s, thus leaving little room for experiment activity.

4. **Disk optimization**
The first bottleneck to tackle is the disk write speed. After a data block is read from a file on tape, it is temporarily stored in the tape server RAM before being sent through the network to a single disk server. The standard disk configuration used at CERN for these disk servers is multiple RAID 1’s. The issue here is that the write speed of a filesystem based on a RAID 1 is more or less equal to the write speed of the slowest of its disks, which in our case varies from 70 MB/s with old machines to 130 MB/s with the top performing machines. The reason why this is a bottleneck is that modern tape drives are instead capable of around 240 MB/s both while reading and writing.

Depending on the size of files being recalled and the size of the tapeserver RAM, multiple streams (up to three) might be used to send multiple files to different filesystems on the same disk server or multiple disk servers. However, this rarely happens, as on average we have 1.1 concurrent streams per recall.

This first bottleneck reduces the recall speed to an average value of around 90 MB/s. In order to remove it, a different RAID configuration is required. Maintaining the redundancy level of the original RAID 1, given that we have to use hardware RAID for maintainability reasons, and given that we need a write speed which would match the read speed of future generations of tape drives, we opted for RAID 10 based on 8 disks (4xRAID 1’s plus 1xRAID 0 on top of them). RAID 10 provides enough disk throughput, up to 400 MB/s, not to hinder the
performance of the tape drive while it is reading. As far as migrations are concerned, RAID 10’s read speeds are twice as high as write speeds, therefore, even more so, tape write performance will not be affected at all. This configuration was also conceived to leave enough headroom for the increasing read/write speed of the next two generations of tape drives.

To ensure that disks will not be a bottleneck during repack operations, one still needs to tackle the problem of concurrent streams accessing the same filesystem. In particular, interleaving reads and writes or onto the same filesystem has obvious negative effects on performance of both operations. Unfortunately there is currently no way in CASTOR to prevent this from happening, as tape streams are assigned to disk servers in a random manner. Therefore the only way of limiting this problem is to make it as unlikely as possible, by using a number of filesystems more or less equal to the number of tapes that are going to be repacked concurrently. Using RAID 10 based on 8 disks for each filesystem instead of RAID 1 based on 2 disks, reduces the number of filesystems available from 12 to 3 per disk server. So in order to maintain the same level of concurrency capabilities, one needs to use four times as many disk servers. In our case we aim at repacking up to 45 tapes concurrently, and tests have shown that 39 disk servers configured each with three RAID 10 filesystems allow all 45 recalls to perform without handicap.

5. Network optimization
Initially, the disk servers dedicated to the repack process were sharing switches with several other general purpose disk servers, and this looked fine. However when these were pumping user data at high rates, our repack disk servers performance was being hindered by the saturated switches. The blocking factor was thus represented by their uplink bandwidth. As the uplinks of these switches were already at their maximum expansion, we needed to change the topology of our repack disk server network so that the disk servers could have their own exclusively dedicated network switches.

Another point for improvement was found during a large scale repack test. Most of the drives were being used at full speed by the repack process. Consequently all tape servers were sending a large amount of data through the network, and the uplinks of the switches they were connected to got saturated as well. Fortunately this problem was simply solved by doubling the uplink bandwidth of the affected switches, given that they were not at their maximum expansion yet.

Tests performed after disk and network configuration changes show that the software had no intrinsic performance limitation besides a small throughput degradation while recalling tapes that contain some files smaller than 30 MB.

6. Improving tape library utilization
Making use of all available drives in the various tape libraries, whilst minimizing interference with ongoing experiment activities, is what completes the performance optimization picture. In the two following subsections we will describe how we achieved this.

6.1. Limiting repack process intrusiveness
Repack is an operations’ activity that needs to be as transparent as possible for the user community. So, the first thing to do is to give user recalls higher priority than repack. This setting makes sure all queued repack recalls get overtaken by user triggered recalls, and this does not represent a long delay for repack, as user recalls usually last only a fraction of a repack recall. This may seem similar to how process scheduling is done by operating systems, however the main difference is that recalls are non preemptive and, when all drives are eventually taken by repack requests, queued user jobs may need to wait for a very long time before getting hold of a drive. Furthermore, CASTOR provides no facility to prioritize migration jobs, therefore user migrations might be queued up behind repack ones.
In order to solve this issue, we dynamically dedicate a number of drives per library to user-only activities. This task is accomplished by a cron job that runs every eight hours, a period long enough to avoid queue length oscillation resonance, and short enough to react promptly to experiment load changes. This script calculates for each library the appropriate percentage of drives to be assigned exclusively to non-repack activities based on the queue length and the percentage of available drives. One important aspect is that, to allow user activity to always sneak in, even when there’s no user queue we always keep 10% of the drives dedicated to user activity, so that under no circumstances the first user job in the queue needs to wait the time of a full repack recall or migration.

6.2. Opportunistic drive usage
Repack needs to be not only transparent, but also efficient. We have mentioned that we have several libraries, and within those libraries there are different types of drives able to read different sets of tapes. The pair <library, drive type> is what we call DGN (Device Group Name), and each DGN has obviously his own queue which cannot be served by any other DGN.

While performing the repack of a large amount of tapes, one has to try to mix the type of tapes in order to leverage the recall load evenly across all DGNs. However this is possible only up to a certain degree, because as we will see in the next section, there are temporal collocation constraints that need to be respected. In addition, user recall activity is obviously DGN agnostic. As far as migrations are concerned, CASTOR poses an additional problem: empty tapes needed for writing are chosen randomly across all available DGNs. The net effect of all these aspects is that in many cases we have some DGNs that are overloaded while some other DGNs are starved.

While we cannot do much for repack recalls, because the set of tapes to be repacked needs to follow a certain order, we can certainly do a wiser choice while deciding on-the-fly which empty tapes to use for repack migrations. We accomplish this through the use of another cron job that chooses the destination tapes based on which DGN has the least load or, when more than one DGN has the same load, randomizes between them. This cron script runs very frequently, i.e. every 45 seconds, to make sure no migration starves due to the lack of empty tapes.

7. Repack status monitoring and temporal collocation
Due to the nature of tape storage, one should limit the number of tape mounts as much as possible. Tape mounts, besides being relatively slow, cause over time wear on both the tape and the drive, and potentially also on the robot itself. That is why it is recommended that, whenever possible, mounts be justified by a sufficient amount of data to be read or written.

In the case of recalls, the ideal approach is to organize data on tapes so that files that are likely to be recalled together are actually put on the same tape. Without trying to be too clever in guessing which files might be recalled at the same time, the philosophy is to place files which were written at the same time in the smallest possible set of tapes. This is ensured by the supply logic of CASTOR tape pools, and it is called temporal collocation.

The repack process can potentially reschedule files around, spreading them on many tapes, causing the temporal collocation to be lost in the destination tapes. We try to limit this effect by repacking at the same time sets of tapes that were originally written in a small time interval. Clearly the smaller the interval, the better the time collocation in the destination tapes.

CASTOR allows one to submit any number of tapes to repack and it guarantees to process them in the same order they were submitted, so care has to be taken when submitting the tapes to make sure their last written dates span a short time interval. To automatize the process we created a database table, called repack status, containing the list of tapes to be repacked. We add tapes to this table with a script that takes a list of tapes as input (a simple text file) and then a cron script periodically takes all the yet unsubmitted tapes from the table and orders them by last written date before actually submitting them for repack. Three more cron jobs
are responsible for the next steps: the first script checks when submitted tapes are actually repacked, the second one moves the repacked tapes to the appropriate delete tape pool where their fate will be decided (either they will be thrown away or reinserted in the system), the third script finally reclaims the tapes in order to make their capacity available for new migrations. Each of these scripts, after having done its job, updates the database table with the current date.

The repack process will be long and can take several months to complete. Monitoring its progress is thus very important to check that everything is going as scheduled. To keep track of all the submitted tapes and their detailed status we use the information contained within the repack status table to periodically produce summary graphs that show how the overall repack process is doing over time.

8. Results

Several repack tests carried out after the repack restructuring have shown considerable speed and efficiency improvements over previous figures. First of all, we increased the recall speed of the latest generation media from 90 MB/s to 205 MB/s (127% increase) and previous generation media from 90 MB/s to 150 MB/s (66% increase), by using RAID 10 setups in the disk servers and by reorganizing the network topology and doubling several switch uplinks. These actions also affected migrations which went from an average of 140 MB/s to 220 MB/s (57% increase).

The global improvement on tape drive throughput over all migrations and recalls is around 80%.

Thanks to the improved throughput, the number of drives needed on average to reach the desired 3.8 GB/s dropped from 43 to 19 for recall (26 for older generation tapes) and from 28 to 18 for migrations. This means an overall reduction of 48% in the number of drives needed on average by the repack process while repacking the latest generation of media and a reduction of 38% while repacking older generation media. This leaves plenty of room for non-repack activities.

We address the typically bumpy experiment activity in two ways: when experiment usage goes up we let it take hold of all the drives it needs by dedicating the majority of them to non-repack activity based on queue length, when experiment usage goes down we make sure we use all free drives available by undedicating drives and by issuing migrations on the least loaded DGNs. In addition, when the libraries are really free, we are able to recall up to 45 tapes concurrently without any speed handicap, thanks to the large amount of disk servers dedicated to the repack process. These improvements allow us to keep a satisfactory overall repack speed average.

The amount of experiment activity on tape drives in 2014 and the availability of new generation tape drives will ultimately determine whether the 115 PB repack exercise will finish within a year or not. However, we now have an efficient infrastructure with very high throughput that potentially allows us to repack all our tapes in around twelve months. Therefore the performance is there and, no matter what the conditions will be, the vital repack activity will be performed in the quickest and least intrusive way possible.

References

[1] NERSC, NERSC Exceeds Reliability Standards With Tape-Based Active Archive
   http://www.activearchive.com/common/pdf/nersc.pdf
[2] Jim Gray, Catharine van Ingen, Empirical Measurements of Disk Failure Rates and Error Rates
   http://arxiv.org/pdf/cs/0701166v1
[3] German Cancio, Vladimir Bahyl, Giuseppe Lo Re, Steven Murray, Eric Cano, Gordon Lee, Victor Kotlyar
   Tape archive challenges when approaching Exabyte-scale
   http://indico3.twgrid.org/indico/getFile.py/access?contribId=280&resId=0&materialId=slides&confId=3
[4] Charles Curran The unbearable slowness of tapes
   http://indico.cern.ch/getFile.py/access?contribId=8&resId=1&materialId=slides&confId=27391