KIT Service Incident Report

Description

On August 19th in 2013, a damaged tape with data of CMS was spotted and the storage experts were tasked to archive as many of the files possibly still cached on disk to tape anew. During these operations, something went wrong and many recent files were not migrated to tape at all. This problem was spotted on October 28th and by that time dCache had already purged 130 files from the disk cache, without having a valid copy on tape for them. Hence, these files are lost.

Impact

130 files are lost for CMS.

Time line of the incident

19.08.2013	Broken tape for CMS discovered.
27.08.2013	Storage experts begin to secure files cached on disk and scheduling them for new migration to tape.
09.09.2013	The migration tasks for saving cached replicas were stopped.
02.10.2013	The experts need to work around a bug in dCache with administrative changes to dCache's pool inventory and databases, in order to progress with the situation.
21.10.2013	The complete situation seemed to be resolved.
28.10.2013	Storage experts notice that a couple of files in CMS' recall queue cannot be fetched from tape, because they were never migrated to tape before. At this point, we found out about the incident and subsequent data loss.

Analysis

Whenever we find a damaged tape, the regular procedure involves the following tasks:

- Finding files from the broken tape, which are still cached on disk.
- Migrating those files to a dCache pool capable of migrating them to tape.
- Report to the experiments in case of data loss.

For these steps we have well established procedures already, which do not bear the risk of data loss – at least under regular circumstances. However, after dCache was updated to version 2.6, a bug was introduced regarding the so-called migration module for the pools. The migration module is used to migrate files internal to dCache in between dCache pools. Such migration jobs have the option to work through a list of file IDs given on initialisation. This option in particular was broken with the dCache version deployed at GridKa at that time: dCache 2.6.5. The migration job did not copy a subset of the files on the pool to its target location, but instead it tried to copy *all* the files from the source to the destination pool (which is the default behaviour, if no filter options are supplied). This issue was noticed on September 9th, were we immediately stopped the migration tasks.

At this point, no real damage was caused. The only downside of what happened was that we had scheduled way more files for archival to tape, then was actually necessary. However, there is another bug in dCache, which caused us trouble now. When flushing files to tape, that had been archived with a dCache version before 1.9.12 already, the new migration is not acknowledged by dCache, although it was carried out successfully. Instead, the pool that performed the migration to tape tries to convince the namespace manager service of dCache (the so-called PnfsManager) to accept the outcome continuously, but is rejected every time. When a certain number of files were migrated to tape this way, the pool does not initialise new migration tasks, because all of its threads are busy arguing with PnfsManager.

Here is where we administrators have to intervene to achieve further progress. First we have to restart the pool service, then we have to manually perform all the actions internal to dCache, that follow upon a regular successful tape migration of a file. If we wouldn't do that, the pool would get into the very same situation again inevitably. It is with these manual changes to dCache internals, that have introduced the wrong status information for 11619 files in total, making dCache believe there would be a valid replica of the files on tape, but in fact the files were never actually written to tape at all.

When we found out about this situation, we acted like we would have spotted another broken tape. I.e. we searched for cached replicas on disk for all of the 11619 files and scheduled a migration job first to another pool and subsequently to tape. Right now, we are still processing many of these files. For 130 files however, we acted too late and dCache had purged them already from disk. 64 more files were already deleted by CMS.

Follow up actions

All the bugs we encountered in dCache were reported to the dCache developers.

- The broken migration module was fixed with the release of dCache 2.6.10 (http://www.dcache.org/downloads/1.9/release-notes-2.6.shtml#10).
- The problem with internal communication about the outcome of successful migration to tape between pools and PnfsManager is not yet fixed with a released version. But the dCache developers confirmed that they will implement a fix for this (support ticket #7192 in the dCache request tracker system).
- Furthermore, the developers acknowledged that handling the incident of a broken tape in dCache may be improved, since it requires manual interventions from the administrators. This topic will be discussed for future releases.
- A couple of more related issues with dCache also have been reported, but are not mentioned within this report in detail anymore.

We also strife to improve our monitoring on the tape system, in order to be able to identify likely hardware failures in advance. In such a case, we have the chance to rescue the data within the tape system and we do not have to rely on luck with dCache (that means, files happen to be cached on disk when a tape breaks).
The list of files lost is send to CMS together with this SIR.

Summary

During a routine operation for rescuing data from a broken tape cartridge, our established procedure was failing due to bugs introduced with the latest golden release of dCache. Adapting to that bug triggered a different bug in the aftermath. Due to this, the time spent to resolve the whole situation was extended a lot and additionally required the administrators to perform changes to dCache's interna, which are out of the ordinary. These internal changes lead to a flawed status of 11619 files for dCache, were dCache believed there would be a valid replica on tape. Actually, for these files no successful migration to tape was accomplished, so we launched another rescue operation for them. For 130 out of the 11619 we acted too late and neither do we have a replica cached on disk, nor on tape anymore.