KIT Service Incident Report

Description

On August 9th 2018, the primary database for CMS' dCache storage element crashed, because there was no space left on its device. GridKa administrators switched to a so-called warm stand-by database within the next hour and production continued for CMS "normally".
However, while the administrators were analysing the problem post mortem the next day, it became apparent, that the stand-by database was lagging behind the master for one week of changes. After a quick discussion with the CMS site contacts, we decided to switch back to the master again. This way, only the work accomplished between August 9th, 14:00 CEST, and August 10th, 11:00 CEST, is lost, instead of everything starting August 2nd.

Impact

319,196 files were extracted from the dCache billing log files, as they were created in the time period when CMS was operating on the backup database system, which was not up-to-date. A list of all those files was handed to the CMS site contacts and about 70k files from /store/unmerged were identified. The majority of the rest were files owned by KIT users.
At the time of writing this document, approximately 58k files were about to be recovered, so a final exact number of lost files cannot be given.

Time line of the incident

August 2nd 2018	Log shipping process from master to slave database started to fail silently.
August 9th 2018, 14:05 CEST	The master database of the CMS dCache storage element crashed.
August 9th 2018, 16:15 CEST	Switch to a so-called warm stand-by database copy completed.
August 10th 2018, 10:00 - 12:00 CEST	Return to the original master database accomplished.

Analysis

Several months ago, GridKa had deployed IPv6 addresses on the dCache nodes serving CMS. However, because of the previously unknown requirement, that IPv6 enabled dCache expects all nodes to be configured IPv4/IPv6 dual-stack, instead of dual-home, the project failed. Not all IPv6 related changes were reverted though; only the most critical IPv6 interfaces were disabled,

leaving those used internally untouched.
By now, GridKa has come up with a new strategy for IPv6 deployment, which we were testing on a pre-production setup, when the legacy IPv6 configuration broke for the slave database node of CMS on August 2nd. This problem went unnoticed, because the tasks that are supposed to replicate the changes from the master to the slave were simply hanging indefinitely, with no timeout. Because those changes are buffered on disk, the partition eventually ran out of space and the master database died.

Follow up actions

There is already monitoring for the synchronisation between master and slave. However, because the actual tasks were not governed by timeouts, they could sleep for days without triggering an error. In order to ensure this situation will not happen again, we have put timeouts on the relevant synchronisation steps.

85k files were found to be cached on any of dCache's disk caches. The complete list was provided to the CMS site contacts, who cut it down to approx. 58k files they would like to recover. GridKa administrators arranged for read-access to a copy of those files via the CMS VO box at GridKa. From there, CMS site contacts have to option to re-import the files into dCache. At the time of writing this document, that process had not concluded yet.

Summary

On August 9th, the database backend for the CMS dCache storage element crashed with no space left on device errors. The administrators quickly moved dCache over to a warm stand-by clone of the database within two hours and all was well for the moment. Only until GridKa administrators discovered the next day, that the stand-by database actually was missing changes performed on the master since August 2nd. Together with the CMS site contacts we decided to switch dCache back to the master database, effectively giving up all work that was accomplished since the database crash the day before. Yet, because no dCache pools were restarted, all file replicas still are accessible on disk, even if the process of locating them is more involved. A list of about 320k files was produced and handed to the CMS contacts. At the time of writing this document, work was ongoing to recover up to 58k of the most important files.

In order to prevent the same situation in the future, the synchronisation tasks between the master and slave databases are now restricted by timeouts. Because that was not the case before, said tasks could sleep for several days and not trigger an error in the monitoring framework. Since the changes to the master database are buffered on disk, the disk space was exhausted eventually, leading directly to the crash of the master database.