CASTOR Name Space Analysis

Massimo Lamanna (18-SEP-2014)

Introduction

In 2014 we performed a thoughtful investigation of the CASTOR name space. The main topics were data security (data on tape organisation as number of copies) and file ownership (coherent file ownership from a Unix point of view and possibility to identify a responsible physical person for each file).

This investigation was needed after recent CASTOR developments, most notably: the introduction of the 'route to tape' (i.e. clear file class to tape pool association); the campaign to increase the number of copies for specific data sets; the drastic reduction of disk-only data and the enforcement of kerberos-based security.

Tape-specific actions (like restoring the number of copy or tape pool reorganisation) are not treated in this document.

In addition this survey is also an opportunity to focus on the core functionality of the CASTOR Name Server and possible evolution.

Glossary

In the CERN instance all files are catalogued under a hierarchy starting with `/castor/cern.ch`. The 3rd level directory `/castor/cern.ch/`**`alice`** is called the **ALICE** top-level directory. We call user directories all the directories under `/castor/cern.ch/user/[a-z]` (for example: `/castor/cern.ch/user/j/johndoe`); user directories of deleted accounts are put in a quarantine under `/castor/cern.ch/user/deleted/[a-z]` and we call them quarantined accounts. The CASTOR Name Space is also referred as NS in the following.

Preliminary actions

Large sets of files (about 30 M files) were still in the CASTOR Name Space although the corresponding files were deleted long ago. During this campaign the name space was cleaned as well: these files were left-overs of stress tests (now the stress tests runs against a copy on the NS), failed tests from PPS, SLS probes, ad-hoc tests from developers and operations, left-overs of disk-only files from retired disk pools. This cleaning put in evidence that now CASTOR is essentially a D0T1 system (D1 data is still reducing and well below the 5% mark).

The ownership of top directories was incomplete. Our policy is that top directories should belong to a responsible (userid) for the computing of the corresponding community and if possible the group should be the corresponding unix group of the owner. For example `/castor/cern.ch/cms` now is owned by `cmscasad:zh` . Most of these cases are now fixed (it was not always possible to have a service account instead of a named person as for old completed experiments: in some cases a CERN staff belonging to the experiment was contacted to serve this purpose).

The name space was scanned to create a dictionary with uid:gid as key to keep the values of number of files and corresponding size. It also reports (for each uid:gid) the NSFILEID of the last file belonging to it (handy for finding example files belonging to a given uid:gid combination). In the preliminary phase this was used to spot suspect combinations like files belonging to a user from one experiment and with a gid from another experiment and non-existing uid:gid combinations. All this can be fixed by "visual" inspection and was generated by erroneous user manipulation or specific tests; the secure access virtually blocked all this for the future; on the other hand, some of this is constantly generated by users changing group (generating "unexpected" uid:gid combinations). The executable is userstats/nameserver_users_stats.py (the corresponding data are collected by a regularly running DB procedure).

User directories: files ownership when the owner account disappears

When an account is blocked, CASTOR does not take any action as this is often a temporary status. At account deletion, the user account is moved under a quarantine tree. Its ACL allow only for the CASTOR operation team to access to these data. For example the tree starting from `/castor/cern.ch/user/j/johndoe` moved onto `/castor/cern.ch/user/deleted/j/johndoe_1234_blocked_timestamp=20140507` with 1234 the UID of "johndoe" before deletion and 20140507 the move time stamp (YYYYMMDD). One should note that as today FIM (the CERN account manager) does keep this information as well (the association between a deleted account, its UID and the information that a given account ever existed all together) but in the past (most) of this info was discarded after deleting the account. To make things potentially confusing some accounts were recycled due to the now removed 16-bit-uid limit.

To set the scale, we observe that daily less than 1‰ are deleted from the system (hence quarantined in CASTOR). A monthly cycle will be enough (scanning each day one username initial) to keep it light and fast enough. This is needed because the browsing of the top of the user directories is rather slow due to (inefficient) LDAP lookups from the nsls command.

The executable is nsstats/quarantineCASTORhomedir.py; it is ran as a daily cron from stager (to use the internal nameserver for performance reasons).

Files ownership when the owner account disappears

All other files (other than user directories) are treated differently. When an account has disappeared the file is reassigned to the owner of the corresponding top directory. For example `/castor/cern.ch/cms/data/johndoefile.dat` would be reassigned (nschown) to the uid:gid pair `cmscasad:zh` following the policy described in the introduction.

During this campaign users have been informed and in some cases other recipes have been applied (several million files under the top directory `/castor/cern.ch/nap` were corresponding to different projects in AB where reassigned as suggested by their coordinators/project leaders).

At variance with the user directories, this activity requires an expensive scan of the entire name space (as selecting all entries where the UID cannot be resolved into a name by LDAP).

Realistically a monthly scan will be enough (it needs an ad-hoc pSQL procedure yet to be defined).

Links (software links)

CASTOR allows to create links (soft links as in `ln -s` using the `nsln` command) with standard Unix semantics. Currently there are about 200k links (less than 1‰ of the catalogue size), mostly in users directories (64%) and in `/castor/cern.ch/grid` (27%). Other important fractions are quarantined users (5%) and under the COMPASS experiment (2%). All the rest is less than 1%.

About 5% of these link files is dangling pointing to CASTOR files not (anymore) known to the system. More interestingly, less than 5% of all links were created after 2010 (and only 1 in 2013!).

A possible deprecation of this feature could start with the end of the distribution of `nsln`. The price of maintaining this feature is actually low and a proper deprecation might be an excessive investment compared to the code simplification (in practice legacy installations would continue to be capable to create links while new distribution won't contain the `nsln` command.

Present status

The number of files of tape is about 270M for a total of 90 PB of data. The NS structure includes about 10% of directories (compared to files).

File sizes is about 300 MB for T1 files; D1 files is about 50 MB (which included the zero-size files).

After the clean up (but before final decommisioning of some D1 areas expected before Run2 start) we observe that the number of D1 files is below 13% but the corresponding size accounts for 2% of the total size catalogued in the NS (double tape copies are counted once in here).

The size is distributed across a small number of power users (the first 10 users accounts for 80% of the total size – see attached table). The 4 LHC experiments plus COMPASS sum up ¾ of the total storage (ATLAS: 33%, CMS: 21%, COMPASS: 15%, ALICE: 10% and LHCb 7%).

NS analysis is performed with nsanalysis/nsanalysis.py and related scripts.

Final considerations on the CASTOR Name Space

1. The CASTOR Name Server is a 300M-entry catalogue using an Oracle DB back-end (operated by the DB group).   The cost of an exit strategy should be evaluated but it is possibly too high due to the fact that its maintenance (both in DB and DSS) is quite low. Since its twin name spaces (LFC and DPM) have been ported to MySQL , it is clear that a similar migration could be done rather quickly.
2. On the other hand the CASTOR Stagers use separate Oracle DB (and in this case the migration won't be simple) there is no compelling reasons to migrate just the NS for now.
3. The NS design was optimised for queries like filename → file metadata (as ownership, checksum, tape location, …) while some operations are extremely painful and expensive (notably listing all files under a given directory requires multiple scans to resolve “all” file objects to decide their position in the file hierarchy. Optimised queries can (and have been) prepared to mitigate this. Since an educated guess on its memory footprint is 150 GB (at 500B per entry) we could profit from the other activities planned on the NS (e.g. VDQM refurbishment) to solve this problem.
4. Some more (accurate) statistics extraction is available now and should be put in production (removing the original scripts from the early days of CASTOR which are known to be broken).
5. Given the present role of CASTOR as an archive system we need a possibility to 'logically remove' a file. In other terms `nsrm` should remove the files from the user view but allow the operation teams to restore it with a simple command. Only repack should remove files irrevocably once the corresponding tape is repacked. Given the current usage of CASTOR (little user activities and the synergy with EOS) it could be that even keeping files after

deletion could be a marginal cost (provided we can completely remove files from specific activities like AFSbackup). The ratio of files (kept/generated) is about 30% across the whole history of CASTOR but it significantly increased in recent years so the impact in the DB will be basically a factor of 2, since one will keep (for a certain amount of time) twice as much entries compared to the present situation.

Previous analyses

https://twiki.cern.ch/twiki/bin/view/CASTORService/IncidentsLostFiles21Apr2010

Attachments

nsanalysis raw output (24-SEP-2014)

```
# Summary (nsanalysis): Wed Sep 24 09:34:45 2014
# Files of tape: 273829056
# Disk-only on files: 34659748
# Directories: 38802991
#
# Top 10 files on tape
#
# na58dst1-vy     48479879 files  (17.7%)
# cmsprod-zh      19259140 files  (7.0%)
# atlast0-zp      14585891 files  (5.3%)
# cms003-zh       13201961 files  (4.8%)
# atltzp1-zp      11020889 files  (4.0%)
# objsrvvy-vy      9772930 files  (3.6%)
# ilc001-zf        8045689 files  (2.9%)
# alicedaq-z2      6177027 files  (2.3%)
# <user>-pz        5967010 files  (2.2%)
# na48cdr-vl       5849871 files  (2.1%)
# The rest is 131468769 files (out of a total 273829056 files: 48.0%)
#
# Top 10 bytes on tape
#
# atlast0-zp      12524752980090236 bytes  (13.7%)
# atltzp1-zp      12170470345621971 bytes  (13.3%)
# phedex-zh       11133150265816809 bytes  (12.2%)
# objsrvvy-vy      8813868737649947 bytes  (9.7%)
# alicedaq-z2      8637191640532069 bytes  (9.5%)
# lhcbprod-z5      6018842976209714 bytes  (6.6%)
# na58dst1-vy      4943025961032096 bytes  (5.4%)
# cms003-zh        4297889298008390 bytes  (4.7%)
# atlascdr-zp      3689110366258297 bytes  (4.0%)
# ilc001-zf        1976818903524748 bytes  (2.2%)
# The rest is 17039906761612106 bytes (out of a total 91245028236356383 bytes:
18.7%)

#
# Top 10 bytes on tape (group)
#
# *-zp      30363902869696648 bytes  (33.3%)
# *-zh      19482456744481869 bytes  (21.4%)
# *-vy      14004053482818763 bytes  (15.3%)
# *-z2       9621478409881985 bytes  (10.5%)
# *-z5       6492014959341900 bytes  (7.1%)
# *-zf       2007776036211774 bytes  (2.2%)
# *-xv       1876356091965615 bytes  (2.1%)
# *-va       1353420512262705 bytes  (1.5%)
# *-za       1213770057828416 bytes  (1.3%)
# *-wj       1120249750381204 bytes  (1.2%)
# The rest is 3709549321485504 bytes (out of a total 91245028236356383 bytes:
4.1%)
#
# Top 10 files on tape (group)
#
# *-vy      62027327 files  (22.7%)
# *-zh      57727725 files  (21.1%)
# *-zp      52690232 files  (19.2%)
# *-z2      13084231 files  (4.8%)
# *-va      12647029 files  (4.6%)
# *-vl       9511524 files  (3.5%)
# *-zf       8858173 files  (3.2%)
# *-z5       8296790 files  (3.0%)
# *-pz       7696899 files  (2.8%)
# *-si       6955941 files  (2.5%)
# The rest is 34333185 files (out of a total 273829056 files: 12.5%)
#
# Percentage of D1 files 12.7%
# Percentage of D1 data size 2.0%
# Mean size of T1 files 333.22 MB
# Mean size of D1 files 53.00 MB
# Percentage of directories over files: 12.6%
```