CASTORFS - A Filesystem To Access CASTOR

Alexander Mazurov*, Niko Neufeld

CERN

(*) corresponding author: alexander.mazurov@cern.ch

1. CASTOR

CASTOR:

- stands for the **C**ERN **A**dvanced **STOR**age manager;
- is a hierarchical storage management (HSM) system developed at CERN used to store physics production files and user files;
- manages disk cache(s) and the data on tertiary storage or tapes.

Files can be stored, listed, retrieved and accessed in CASTOR using command line tools or applications built on top of the different data transfer protocols.

Example of using command line tools:

rfdir `/castor/cern.ch`
- List directory contents

rfcp `/castor/path/to/file /tmp`
- Copy files from CASTOR to local filesystem

2. CASTOR filesystem

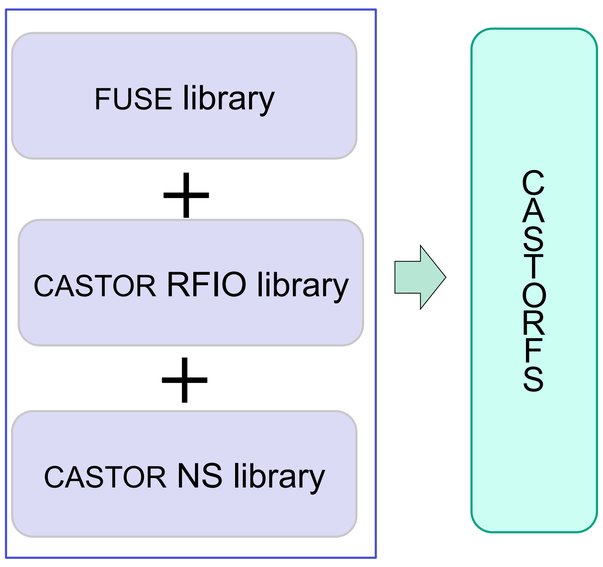
CASTOR logically presents files in a UNIX like directory hierarchy of file names. This suggests to implement a new filesystem capable of operating on files stored on CASTOR using standard Unix operation system calls and commands like open, read, cp, rm, mkdir, ls, cat and find.

We have implemented a complete POSIX filesystem to access CASTOR using **FUSE** (Filesystem in Userspace) and have successfully tested and used this on SLC4 and SLC5 (both in 32 and 64 bit).

We call it **CASTORFS**.

3. Implementation

The FUSE library is the main component of **CASTORFS**. Using it with the CASTOR **RFIO** (Remote File I/O) and CASTOR **NS** (Name Server) libraries allows implementing a fully functional filesystem in a userspace program.



The CASTOR RFIO library provides the main input/ output operations for files in CASTOR: rfio_readdir, rfio_open, rfio_read, rfio_mkdir, rfio_stat,… – each operation has a corresponding POSIX operation: readdir, open, read, mkdir, stat,….

The CASTOR NS library provides information about files and directories in CASTOR. The library not only allows to retrieve standard meta-data like creation and modification time, owner and size of a file/directory, but also to get some additional attributes, such as checksum of a file, status of a file with respect to migration from disk cache to tape.

4. Usage

An example of using CASTORFS, which mounts a new filesystem to a /castorfs local directory:

```
$> /opt/castorfs/bin/castorfs /
castorfs  -o allow_other,
castor_uid=10446,
castor_gid=1470,
castor_user=lbtbsupp,
castor_stage_host=castorlhcb,
castor_stage_svcclass=lhcbraw
```

From command line we can set up:
- connection parameters for CASTOR
- Information about the CASTOR user
- Mode of the connection: read only or full access

After mounting a CASTORFS filesystem you can run any of your favorite tools intended to work with files or directories:

```
$> ls –la /castorfs/path/to/file
$> mkdir –p /castorfs/path/to/dir
$> cat /castorfs/path/to/file
$> rm /castorfs/path/to/file
$> find /castorfs | grep “strangefilename.raw”
```

5. Extended attributes

The CASTOR name server stores some interesting file meta-data, for example: a file checksum, a status of migration to tapes (migrated or not).

In CASTORFS we implemented a mapping of additional file information to filesystem extended attributes:

```
$> getfattr -d /castorfs/cern.ch/lhcb/online/data.tar
# file: castorfs/cern.ch/lhcb/online/np.tar
user.checksum="569145875"
user.checksum_name="adler32"
user.nbseg="1"
user.status="migrated"
```

6. Packaging

CASTORFS is available as a RPM package for Scientific Linux CERN (SLC4 and SLC5, both 32 and 64-bit architectures)

In the LHCb Online cluster (CERN) the CASTORFS RPM has been distributed to the compute farm using Quattor/LinuxFC (a system administration toolkit for automated installation, configuration and management of Linux-clusters).

If you have the same user (UID, GID) on the client host as she is registered with CASTOR, you can use the FUSE and CASTOR packages provided by their vendors. If you want to map your local user to a ***different*** user on CASTOR you need to apply our patches to the FUSE and CASTOR libraries.
These patches will be obsolete once CASTOR can use certificates.

7. Performance

Due to some limitation in the Linux kernels prior 2.6.27, we have a performance problem for writing and reading files to/from CASTOR compared to native RFIO about 5 times slower for writing and 2 times slower for reading(*).

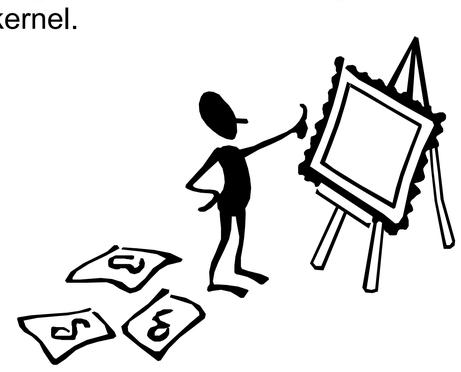
When testing CASTORFS with the latest kernels we observe significantly better performance.

(*) This is for the first read, subsequent reads are much faster, as they go through the buffer-cache

8. TODO

An improvement of read- and write performance is the main item on the TODO-list:

- Implementation of a caching mechanism for storing information about frequently requested CASTOR file meta-data.
- Creating a CASTORFS package for the newest Linux kernel.



9. Useful resources

- **CASTORFS web page**
  - https://lbtwiki.cern.ch/bin/view/Online/CastorFS
- **CASTOR web site**
  - http://cern.ch/castor
- **FUSE web site**
  - http://fuse.sourceforge.net
- **Quattor web site**
  - http://apps.sourceforge.net/mediawiki/quattor/index.php?title=Main_Page