xCastor2 xrootd and rtcpd

possible implementation (sep 2009)

Victor Kotlyar

What is rtcpd

- is a main daemon to handle data transfers between tape and diskserver.
- Pure C code.
- Was written in 2000 for CASTOR1 and is backward compatible with SHIFT (has some not used code anymore?).
- Fully multithreaded.
- Uses rfio API.
- Makes tape part fully independent of stager/nameserver and can be used as tape implementation for other systems.
- Have 9years of usage history all features/bugs well known.

What is rfiod

- Is a main daemon to handle rfio data transfer protocol which is **CORE** of whole CASTOR system in using diskserves.
- Stand alone multithreaded daemon which provide POSIX-like access and has additional function (not POSIX).
- Has history from early 90's starting from SHIFT, was improved with data streaming mode for CASTOR1.
- rfio API is very well documented and has been used for a long time with a lot of people and experiments.
- Transfers speed in data streaming mode is the same as for the others "good" protocols.

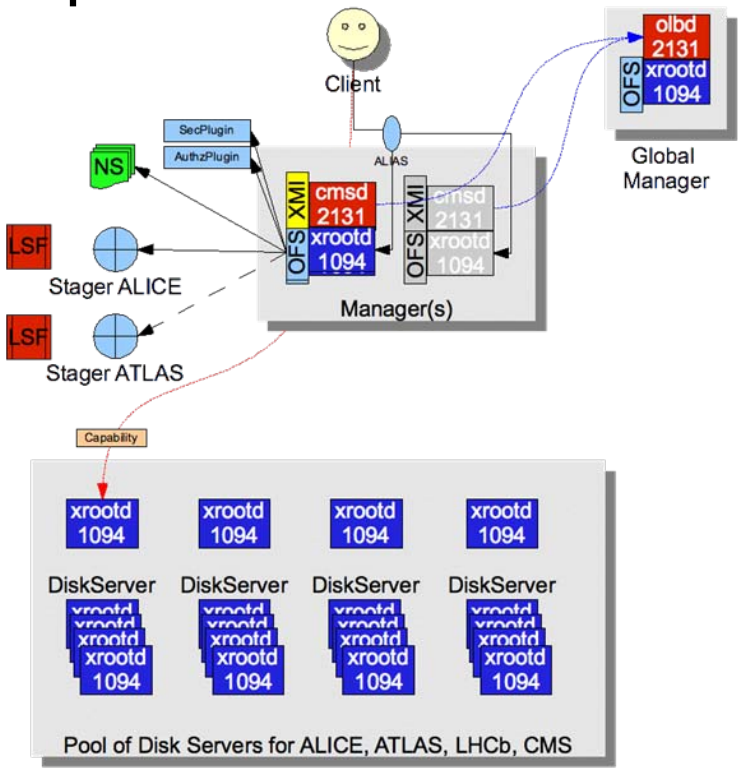
What is xrootd

- The xrootd server is designed to provide POSIX-like access to files and their enclosing directory namespace.
- The project of Stanford Linear Accelerator Center started in 2003, C++, multithreaded, allow to build Scalable Cluster Architecture for Low Latency Access (has xrootd managers/redirectors and global redirectors).
- Based on a run-time plug-in mechanism so that new features can be added with a minimum of disruption.
- Xroot C++ API for client: XrdClient, XrdPosixXrootd (based on XrdClient). C XrdPosix wrapper to XrdPosixXrootd. There is no a good documentation for API (need to dig into the sources).

What is xCastor2

- 2nd generation of an xrootd interface to Castor2.
- A manager's part and a diskserver's part.

• A client connects and authenticates to a manager node.
• The manager authorizes the client access using the castor nameserver.
• The manager applies policies to each request to define the location of a file either via a cache or castor services.
• The disk server verifies for every open the manager signature for the requested file and allows or denies access to a physical file.
• Disk server are passive and don't subscribe to manager nodes.



What to implement in rtcpd

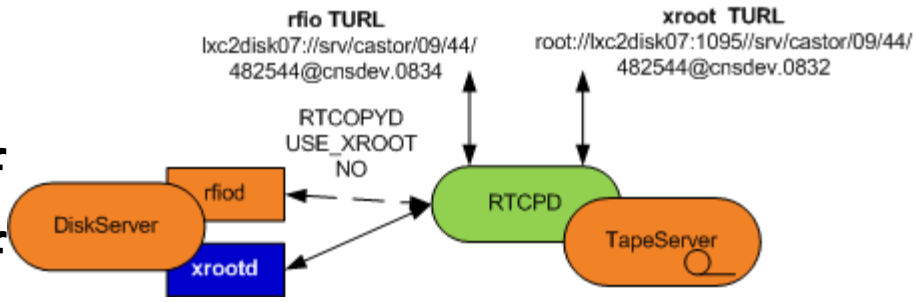
- Limited set of rfio calls, no need to change working logic.
- POSIX-like calls:
  rfio_mstat64, rfio_end, rfio_open64, rfio_lseek64, rfio_read, rfio_write, rfio_close, rfio_serror.
  Could be implement with xroot POSIX-like API.
- Non POSIX-like calls (FORTRAN sequential file access):
  rfio_xyopen, rfio_xysock, rfio_xyread, rfio_xywrite, rfio_xyclose.
  Are rfio protocol specific and no need anymore (tapegateway do not care about them).
- There are no special case for open (mode bits or something other).
- rfio_read/write have the same logic as xrootRead/Write.
- We can use POSIX errno to handle errors (rfio_serror can work with it too).

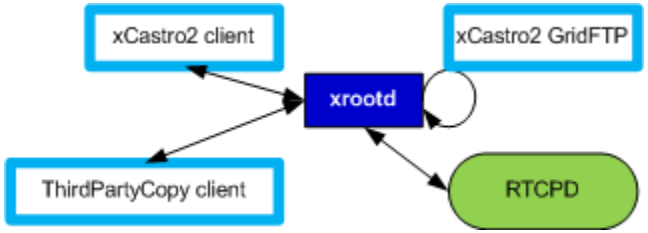
How to implement

- Add replacement for rfio POSIX calls with XrdPosixXrood interface calls.
- Make a C++ code from rtpc_CheckReq.c, rtcpd_Disk.c to be able to use xroot API.
- Add additional logic to handle rfio to xroot tURL conversion and recognize xroot tURL (use modified xroot tURL for xCastor2 xrootd).
- There is no need to change whole logic in rtcpd and we can use error handling based on POSIX errors.
- Need a modification of a filesystem plug-in and an authorization plug-in for xrootd castor for diskservers: XrdxCastor2Ofs.cc, XrdxCastor2ServerAcc.cc (to allow direct access from tape servers and correct usage of stat, rewrite commands).

What we have

- Rtcpd can handle rfio and xroot request simultaneously.
- We can enable converting of all rfio requests to xroot (if we have xrootd installed on the same diskservers)
- We can use advanced authentication/authorization of xrootd (not only unix uid/gid and there is a special authorization plug-in)
- We can use bandwidth limits for different parts of system (right now for each stream, but it is possible to modify the rate limiter in many ways)





How it possible to use it at present

- To add to the current implementation of xCastor2 to use xroot as much as possible.
- To use a new rtcpd with a tape aggregator who will be able to provide xroot tURL.