Ins & Outs of XRootD POSIX Layer

XRootD Workshop

UCSD

January 28, 2015

Andrew Hanushevsky, SLAC

http://xrootd.org

XRootD POSIX Interface

- Provides a well known interface
  - Allows mixture of local and remote files
    - All in a single application based on file path
  - Trivial migration to other storage types
- Only synchronous operations supported
  - An asynchronous open() is available
    - This is a non-standard extension
  - 52 standard POSIX functions supported

Programmatic Interface (C/C++)

- #include "XrdPosix/XrdPosixExtern.hh"
  - Compile with "-I /usr/include/xrootd"
- Call XrdPosix_*xxxx*(…)
  - Where *xxxx* is the POSIX "C" function
    - First letter capitalized
  - E.G. open(…) becomes XrdPosix_Open(…)
- Link with "-l XrdPosix"
  - Uses libXrdPosix.so

Programmatic Requirement

- You must define one global POSIX object
  - This can be a static or allocated before use
- #include "XrdPosix/XrdPosixXrootd.hh"
- XrdPosixXrootd myFS(*maxfd*);
  - You can also use myFS methods
    - However, these only are for remote access
    - Path must be a valid URL

What is *maxfd*?

- XrdPosixXrootd myFS(*maxfd*);
  - *maxfd* sets max number of open file descriptors
  - Only applies to remote files not local ones
    - Used to allocate a file descriptor table plus…
    - XrdPosix FD's are shadowed by real FD's
      - This keeps the program honest & detects errors
        - The XrdPosix returned FD points to /dev/null
    - If *maxfd* is negative, FD shadowing is not used
      - The FD is internal and higher than the hard limit
      - Performance option for honest unfettered programs

How It Works

- Calls to XrdPosix_xxxx(*path*,…)
  - Call vectored to POSIX object if *path* is
    - root://…
    - xroot://…
    - Matches a virtual mount point (VMP)
  - Otherwise, call vectored to local POSIX i/f
- Calls to XrdPosix_xxxx(*fd*,…)
  - Simple to determine if xroot or local *fd*

Virtual Mount Points

- Automatically converts a path to "xroot://"
  - Controlled by XROOTD_VMP envar
    - XROOTD_VMP = *server*[:*port*]:<*path*>[=[*newpath*]]
      - Can have multiple specs each separated by a space
        - Path prefixes are matched in right to left order
      - If *newpath* missing the prefix is stripped
      - If "=" is missing path is not changed
      - If *port* is missing it defaults to 1094
    - Paths prefixes not matching *path* are untouched

Virtual Mount Point Example

- XROOTD_VMP=srv.domain.edu/xrootd/=/atlas/
  - Path like "/xrootd/myfile" internally becomes
    - "xroot://srv.domain.edu//atlas/myfile"
    - Which is vectored to xrootd
  - Path like "/tmp/myfile" is untouched
    - Which is vectored to local POSIX interface

Using Unmodified Applications

- Done via a preload library
  - Wrapper script on an rpm installed node
    - LD_PRELOAD=libXrdPosixPreload.so
    - export LD_PRELOAD
    - $*
  - Works best with XROOTD_VMP envar
    - Avoids application problems with URL-like paths
      - Apps using unsupported POSIX calls will not work!

Acknowledgements

- Current Software Contributors
  - ATLAS: Doug Benjamin, Patrick McGuigan, Ilija Vukotic
  - CERN: Lukasz Janyst, Andreas Peters, Sebastien Ponce, Elvin Sindrilaru
  - Fermi: Tony Johnson
  - Root: Gerri Ganis
  - SLAC: Andrew Hanushevsky, Wilko Kroeger, Daniel Wang, Wei Yang
  - UCSD: Alja Mrak-Tadel , Matevz Tadel
  - UNL: Brian Bockelman
  - WLCG: Mattias Ellert, Fabrizio Furano, David Smith
- US Department of Energy
  - Contract DE-AC02-76SF00515 with Stanford University

Now For The Demo!

SLAC NATIONAL ACCELERATOR LABORATORY