Xrootd Preload Library

The xrootd preload library

The xrootd preload library allows standard POSIX I/O calls to xrootd servered files in stead of calls to the local file system.

Using the preload library

The script xrdprel should be used to run a command using the xrootd preload library. The script sets the proper environment variables (LD_LIBRARY_PATH, LD_PRELOAD) and then executes the provided command. The LD_PRELOAD variable instructs the loader to load an additional library in this case the xrootd preload library. Certain function calls (e.g.: open, close, read, fopen, fclose, fread, ...) are called from the preload library which if a file is a xrootd file will subsequently use the xrootd tools to access the file. If a file is not and xrootd file the call will be redirected to the corresponding system call.

The usage is:

Usage

`xrdprel [-g] [-h]  _command_ _args_`
 	
`_command_ and _args_ are the user command and and its arguments`
 	
`-h: prints a help`
`-g: use a virtual mountpoint, which means file names starting with /glast/ are assumed to be`
`    xrootd files.`

The xrdprel is part of the xrootd client tools:

> */afs/slac.stanford.edu/g/glast/applications/xrootd/PROD/bin*

> **Listing directories**
>
> Listing a directory with xrdprel (e.g.: xrdprel -g ls /glast/) will fail as the command is handled by the redirector which has no direct access to the data file systems. Issuing the command against a data server (e.g.: xrdprel ls root://wain084//glast/) will work but only show the listing from that single server. To get a listing from all servers use the **xrdls** command.

Examples:

ls a file:

> *xrdprel ls -l*

Would run ls against a file in xrootd.

ls a file using virtual mount point:

> *xrdprel -g ls -l /glast/admin/mon_all.tst*

This has the same effect as the previous example. The name /glast/admin/mon_all.tst will automatically resolve to root://glast-rdr//glast/admin/mon_all.tst

Read a fits file

> *xrdprel -g fverify /glast/mc/ServiceChallenge/obssim_v9r5/ft1/obssim_v9r5-000000_events_0000.fits*

The -g option and /glast/... file names must be used with fitsio tools.

Issues (Important)

32/64 bit version

The main issue is loading the proper preload library. A 32 bit binary needs the 32 bit preload lib whereas a 64 bit compilied executable needs a 64 bit version. Right now by default always the 32bit preload library is used, which will cause a failure if running a 64 bit application on a 64 bit OS. The application will still run but the preload library will not be loaded.

Also, one has to be careful if 32 and 64 bit application are mixed. For example calling **fverify** (32bits) from within **python** (64 bits on rhel4-64).

Reading fits files, xrootd:// prefix

The cfitsio package works with the xrootd preload library. However, cfitsio decides how to access files depending on the prefix. Files starting with root:// will be open with the rootd (not xrootd) protocol which is not supported by the GLAST xrootd. Therefore, the file names as obtained by the data catalog, have to be stripped of the rootd://glast-rdr/ prefix and the **-g** option must be used for xrdprel:

> *xrdprel -g fverify /glast/mc/ServiceChallenge/obssim_v9r5/ft1/obssim_v9r5-000000_events_0000.fits*