



Michal Simon

Metalink & XRootD



Outline

- Introduction
- Virtual Redirector
- Extreme copy
- Knobs and buttons

Introduction

- XRootD client supports **both metalink 3.0 and 4.0**
- XRootD client can interact with **both remote and local metalink files**
 - Local metalink file can be obtained e.g. by
`rucio list-file-replicas -metalink`
- Currently XRootD client honors only `root/xroot` replicas within the metalink

Virtual Redirector

- Metalink file is first downloaded (if not local) and then parsed
- Based on the Metalink content a **virtual redirector** is being created (resides at the abstraction level of a `XrdCl::Channel`)
- Currently Metalinks are the only virtual redirectors in XRootD client

Virtual Redirector

- Acts as a **meta-manger** (the fact it is metalink-based is transparent to the user)
- Generates **virtual redirect responses** and if necessary error responses
- Can be also used to **query file size and checksum** (if available in the Metalink file)

Introduction

- Takes into account `tried` opaque info when selecting a replica
- As meta-managers, **Metalinks can recover some additional errors:**
 - `kXR_Unsupported`
 - `kXR_FileLocked`
 - `kXR_NotAuthorized` (by default 3 times, can be tuned using `XRD_NOTAUTHORIZEDRETRYLIMIT`)

Extreme Copy

- A metalink can be used by `xrdcp` to do multi-source file transfer (**BitTorrent** like)
- User just needs to specify the number of sources to be used and a Metalink file as a source
- `xrdcp -y 4 root://host//dir/file.meta4 .`

Knobs and buttons

- XRootD client will ignore a `kXR_wait` response if aggregated wait time for given request exceeds configured maximum (by default 60s)
- Can be tuned using `XRD_MAXMETALINKWAIT`

Knobs and buttons

- Metalink processing can be turned off if the user wishes to download the actual metalink file
- Can be turned off by setting `XRD_METALINKPROCESSING` to 0

Knobs and buttons

- When Metalink support was introduced, XRootD client did not support yet local files
- To allow local Metalinks at that time *localfile* semantics has been introduced
 - `root://localfile//path/to/local.meta4`
- This semantics is deprecated, however can be enabled by `XRD_LOCALMETALINKFILE` envvar (will generate a warning)

Knobs and buttons

- A GLFN redirector is a redirector of last resort
- The GLFN tag has to be specified in the Metalink
- The redirector has to be specified using `XRD_GLFNREDIRECTOR` `envar`

Knobs and buttons

- Not-authorized error can be recovered at Metalink by default 3 times
- This can be tuned using: `XRD_NOTAUTHORIZEDRETRYLIMIT`

Multi protocol Metalink

- Introducing channel-level plug-ins would make it possible to support multi protocol Metalinks (e.g. xroot/root and HTTP)
- Channel level plug-ins
 - A plug-in that resides on the **same level as virtual redirector** (Metalink)
 - Would allow for **redirecting requests to protocol plug-ins** (e.g. HTTP)
 - Planned for 5.0.0 (or 5.0.1)

Questions?

