



Michal Simon

XRootD client developments



Outline

- XRootD Introduction
- New features
 - Metalink support
 - Extreme Copy
 - ZIP archive support
 - Signed Requests
- Plans for 2017

XRootD Introduction

- Backbone of EOS
- Two developer actively working on XRootD client
- Release management

Metalink support

- Available through xrdcp and XrdCl::File API
- Support for both metalink 3.0 and 4.0
- Enabled by default (can be switched off using XRD_METALINKPROCESSING)
- xrdcp supports local metalink files
(by convention: root://localfile//path/metalink)

Metalink, selecting source

- Virtual redirector (familiar cmsd experience)
- Metalink priorities determine the order
- If there are no more replicas to try, the GLFN redirector of last resort kicks in if:
 - GLFN tag is specified in the metalink
 - XRD_GLFNREDIRECTOR is set

Metalink, others ...

- If target filename is omitted from the CLI the name attribute from file tag is used as the target file name
- If the requested checksum is present in the metalink it is used as the server checksum
- metauris are NOT supported

Extreme Copy support

- Supported through xrdcp and XrdCl::CopyProcess API
- User specifies only the number of sources
- The data servers are determined using deep locate or a metalink file

Extreme Copy: the algorithm

- The file is being partitioned into chunks
 - not too small so the sources benefit from sequential read
 - not too big so the destination is not overwhelmed with large sparse files
 - Tunable through an env var
- Fast sources are allowed to steal work from slow ones

ZIP archive support

- Supported through xrdcp, API might be exposed in the future when stable
- -z option allows to specify a file name that should be extracted from a ZIP archive
- Use case: extract root files from ZIP archive (no decompression)

ZIP archive: implementation

- Client checks the offset of the requested file in the ZIP archive's Central Directory record
- The file is being read starting at respective offset (pure client side implementation)
- Due to poor layout of a ZIP archive the last 64KB of the archive have to be read (archives $\leq 64\text{KB}$ are downloaded entirely)
- Under construction: check-summing, archive listing, possibly decompression

Signed Request

- During handshake the server specifies the required protection level
- The sha2 algorithm is being used to compute the hash of the request
- Afterwards, the symmetric key (session specific) is being used to encrypt the hash
- Each signature has a sequence number that prevents replay attacks

Signed Request: compatibility

- Backwards compatibility:
 - The client only signs the request if instructed so by the server
 - ‘compatible’ security level requires the client to sign only potentially destructive requests (old clients are mostly used for read-only data access)

Plans for 2017

- Proxy delegation + pull / push request
- Dynamic source selection
- Support local files in the XRootD client
- Redirect backtrace

Questions?