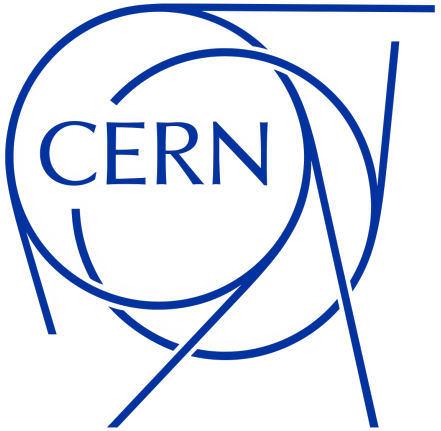


XRootD - HTTP TPC Client support XrdClHttp

Diogo Mattioli

September 2026

Introduction

Making HTTP-TPC a first class citizen in the XRootD client

XrdClCopy and XrdClHttp

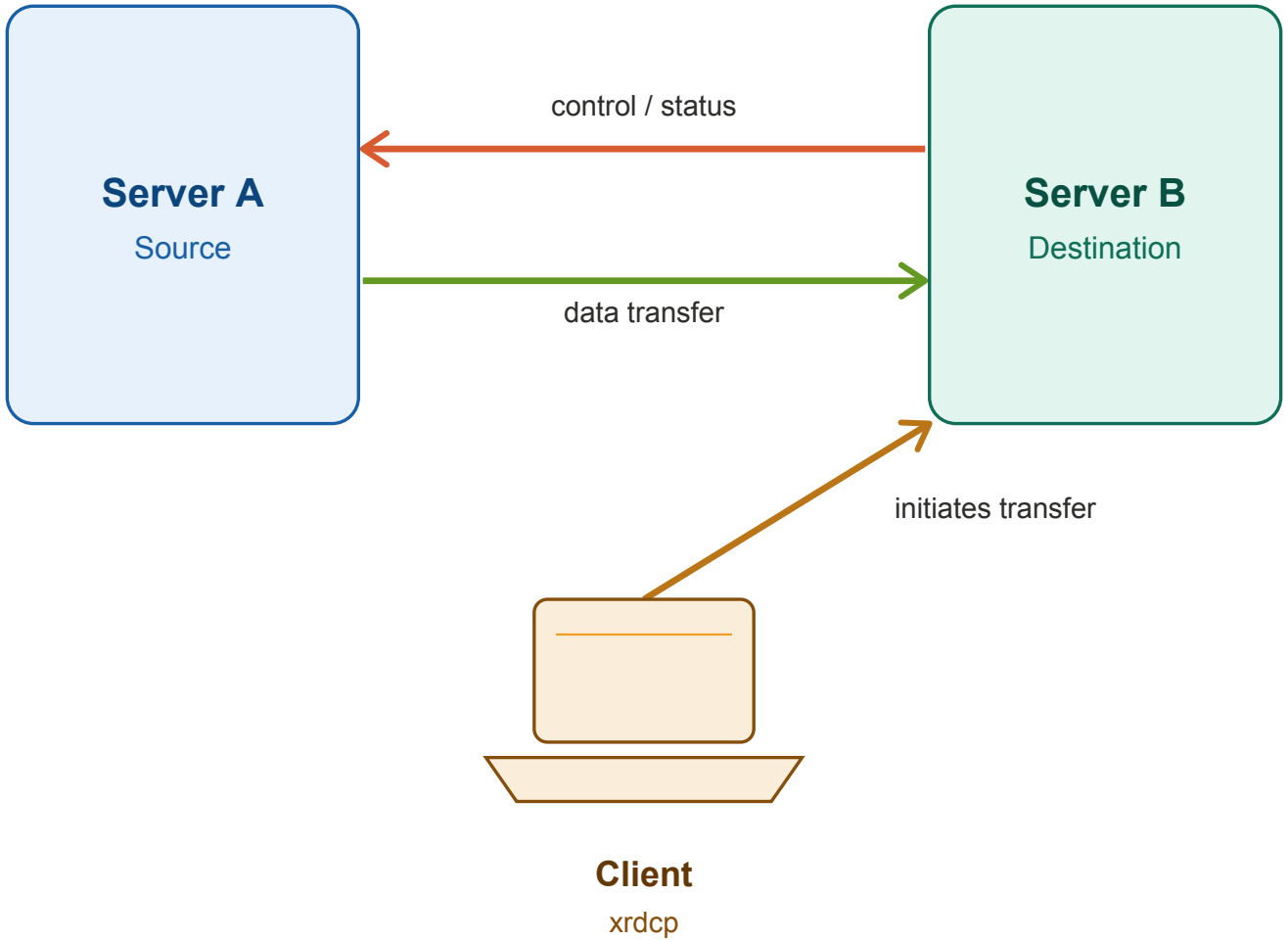
The libdavix → libcurl refactor modernized XRootD's HTTP client while exposing feature gaps between XrdClHttp and XrdClCopy. This work bridges the two, enabling HTTP transfers through the copy infrastructure and laying the groundwork for Third-Party Copy (TPC).

- Absorbing Davix use cases

Motivation

- Connecting the dots between XrdClHttp and XrdClCopy
- Interface support for Third-Party Copy (TPC), as well as other protocols
- Additional requirements, including:
  - Authorization / Token files
  - Pull vs. push selection
  - Multi-stream support
  - Other related capabilities

How it works


Server A
Source
control / status
data transfer
Server B
Destination
initiates transfer
Client
xrdcp

Features

HTTP Third-Party Copy (TPC)

- Execute HTTP TPC copy jobs

Interface support for TPC

- Allow implementation of TPC for other protocols

Progress Tracking

- PerfMarker feedback integrated into the copy job and progress bar

Multistream Transfers

- Support for parallel streams via X-Number-Of-Streams

Overwriting Files

- —force injects Overwrite header

Push & Pull Copies

- -t push enables PUSH copies; PULL is the default mode

TransferHeader Manipulation

- Dynamically set TransferHeaderAuthorization according to PULL or PUSH direction

Token File Support

- --tpc-token-file option to specify the token file (BEARER_TOKEN_FILE)

Features (Interface support for Third-Party Copy)

XrdCl/XrdClThirdPartyCopyPlugIn.hh

```
class ThirdPartyCopyPlugIn
{
  public:
    virtual ~ThirdPartyCopyPlugIn();

    //------------------------------------------------------------------------
    //! Third party copy
    //!
    //! The copy runs to completion before this method returns.
    //!
    //! @param source           the file to be copied from
    //! @param dest             the location the file is copied to
    //! @param properties       properties of the copy, none if null
    //! @param progress_handler handler notified while the copy runs, none if
    //!                         null
    //! @param timeout          timeout value, if 0 the environment default
    //!                         is used
    //! @return                 status of the operation
    //------------------------------------------------------------------------
    virtual XRootDStatus ThirdPartyCopy( const std::string  &source,
                                         const std::string  &dest,
                                         const PropertyList *properties,
                                         ProgressHandler    *progress_handler,
                                         time_t              timeout ) = 0;
};
```

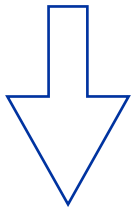
XrdClHttp/XrdClHttpFilesystem.hh

```
#include <XrdCl/XrdClThirdPartyCopyPlugIn.hh>

class Filesystem final : public XrdCl::FileSystemPlugIn,
                         public XrdCl::ThirdPartyCopyPlugIn {
public:
    virtual XrdCl::XRootDStatus ThirdPartyCopy( const std::string           &source,
                                                const std::string           &dest,
                                                const XrdCl::PropertyList   *properties,
                                                XrdCl::ProgressHandler      *progress_handler,
                                                time_t                       timeout = 0 ) override;

}
```

http://, xyz://, etc.

Features (Progress Tracking)

```
Perf Marker\n
Timestamp: 1537788010\n
Stripe Index: 0\n
Stripe Bytes Transferred: 238745\n
Total Stripe Count: 1\n
RemoteConnections: tcp:129.93.3.4:1234,tcp[2600:900:6:1301:268a:7ff:fef6:a590]:2345\n
End\n
```

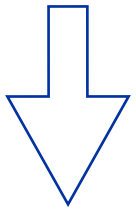


```
$ xrdcp -T only -f http://localhost:1094//sourcefile http://localhost:1095//destinationfile
[96.97GB/273.7GB][ 35%][=================>                                  ][6.465GB/s]
```

https://twiki.cern.ch/twiki/bin/view/LCG/HttpTpcTechnical

Features (Multistream Transfers)

```
$ xrdcp --streams 10 -T only -f http://localhost:1094//sourcefile http://localhost:1095//destinationfile
[18.68GB/273.7GB][  6%][===>                                                ][1.868GB/s]
```

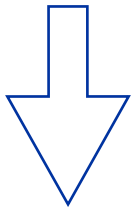


```
260907 09:44:54 11286 http_Protocol:  rc:50 got hdr line: COPY //destinationfile?xrdcl.intent=tpc HTTP/1.1
260907 09:44:54 11286 http_Protocol:  Parsing first line: COPY //destinationfile?xrdcl.intent=tpc HTTP/1.1
260907 09:44:54 11286 http_Protocol:  rc:22 got hdr line: Host: localhost:1095
260907 09:44:54 11286 http_Protocol:  rc:45 got hdr line: User-Agent: xrdcl-http/v6.1.1-85-g3904ad4be
260907 09:44:54 11286 http_Protocol:  rc:13 got hdr line: Accept: */*
260907 09:44:54 11286 http_Protocol:  rc:60 got hdr line: Source: http://localhost:1094//sourcefile?xrdcl.intent=tpc
260907 09:44:54 11286 http_Protocol:  rc:25 got hdr line: X-Number-Of-Streams: 11
260907 09:44:54 11286 http_Protocol:  rc:14 got hdr line: Overwrite: T
```

https://twiki.cern.ch/twiki/bin/view/LCG/HttpTpcTechnical

Features (Overwriting Files)

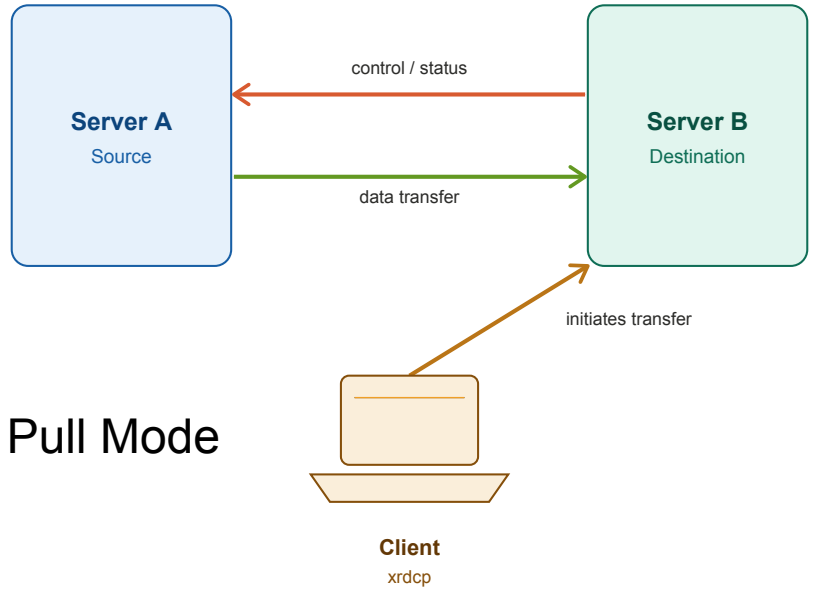
```
$ xrdcp -S 10 -T only --force http://localhost:1094//sourcefile http://localhost:1095//destinationfile
[18.68GB/273.7GB][  6%][===>                                                    ][1.868GB/s]
```



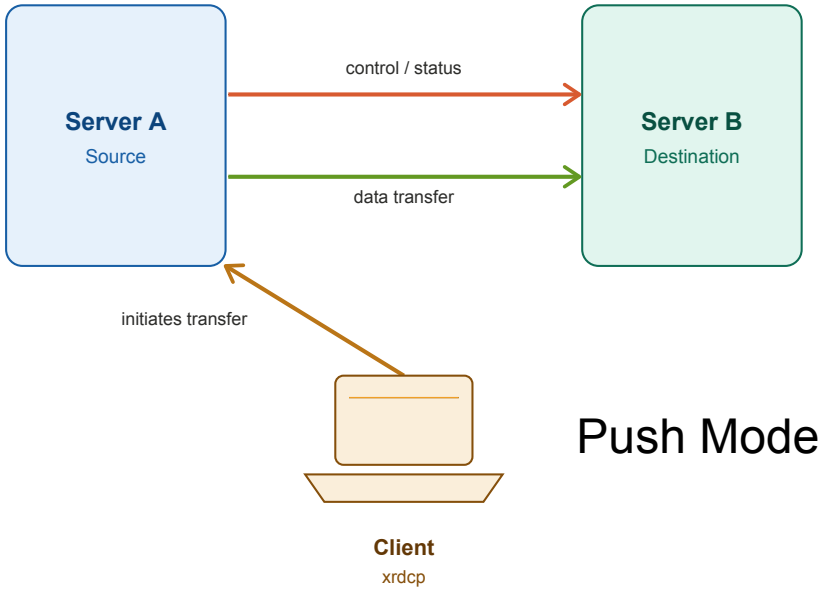
```
260907 09:44:54 11286 http_Protocol:  rc:50 got hdr line: COPY //destinationfile?xrdcl.intent=tpc HTTP/1.1
260907 09:44:54 11286 http_Protocol:  Parsing first line: COPY //destinationfile?xrdcl.intent=tpc HTTP/1.1
260907 09:44:54 11286 http_Protocol:  rc:22 got hdr line: Host: localhost:1095
260907 09:44:54 11286 http_Protocol:  rc:45 got hdr line: User-Agent: xrdcl-http/v6.1.1-85-g3904ad4be
260907 09:44:54 11286 http_Protocol:  rc:13 got hdr line: Accept: */*
260907 09:44:54 11286 http_Protocol:  rc:60 got hdr line: Source: http://localhost:1094//sourcefile?xrdcl.intent=tpc
260907 09:44:54 11286 http_Protocol:  rc:25 got hdr line: X-Number-Of-Streams: 11
260907 09:44:54 11286 http_Protocol:  rc:14 got hdr line: Overwrite: T
```

https://twiki.cern.ch/twiki/bin/view/LCG/HttpTpcTechnical

Features (Push & Pull Copies)



Pull Mode

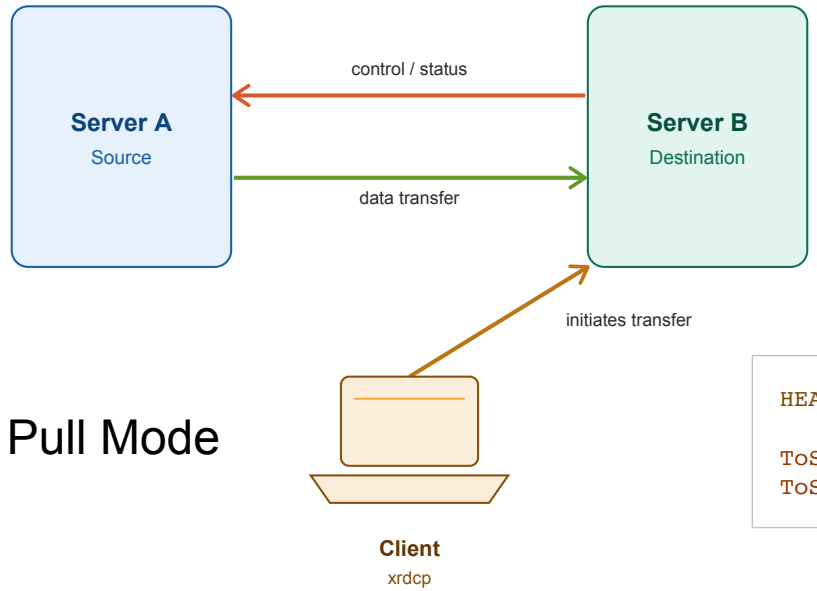


Push Mode

```
$ xrdcp -T only -f http://localhost:1094//sourcefile http://localhost:1095//destinationfile
[96.97GB/273.7GB][ 35%][=================>                                  ][6.465GB/s]

$ xrdcp -T push only -f http://localhost:1094//sourcefile http://localhost:1095//destinationfile
[96.97GB/273.7GB][ 35%][=================>                                  ][6.465GB/s]
```

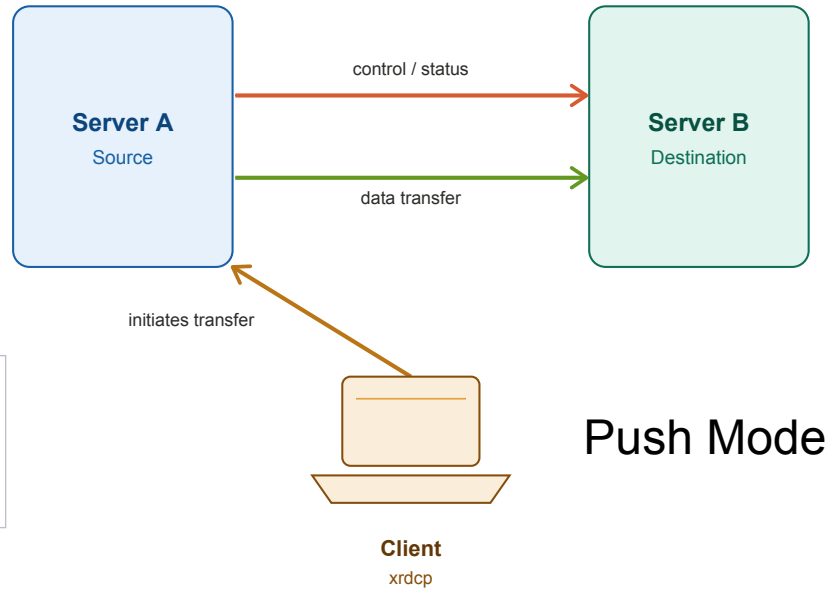
https://twiki.cern.ch/twiki/bin/view/LCG/HttpTpcTechnical

Features (TransferHeader Manipulation)



Pull Mode

```
HEADERS

ToServerA: Header Content A
ToServerB: Header Content B
```



Push Mode

```
COPY /destination/my-data.root HTTP/1.1
Host: B
Source: https://A/source/path/my-data.root
ToServerB: Header Content B
TransferHeaderToServerA: Header Content A
```

Where client connects

```
GET /source/my-data.root HTTP/1.1
Host: A
ToServerA: Header Content A
```

Where XRootD server connects

```
COPY /source/my-data.root HTTP/1.1
Host: A
Destination: https://B/destination/my-data.root
ToServerA: Header Content A
TransferHeaderToServerB: Header Content B
```

Where client connects

```
PUT /destination/my-data.root HTTP/1.1
Host: B
ToServerB: Header Content B
```

Where XRootD server connects

https://twiki.cern.ch/twiki/bin/view/LCG/HttpTpcTechnical

Features (Token File Support --tpc-token-file)

```
$ xrdcp -T only --tpc-token-file /path/to/token -f https://A//file https://B//file
$ xrdcp -T push only --tpc-token-file /path/to/token -f https://A//file https://B//file
```

Syntactic sugar to inject ‘Authorization’ and ‘TransferHeaderAuthorization’ tokens

Plain

```
Token A
Token B
```

```
Token A
```

```
[newline]
Token B
```

JSON

```
{
    "src": "Token A",
    "dst": "Token B"
}
```

```
{
    "src": "Token A"
}
```

```
{
   "dst": "Token B"
}
```

Environment variable BEARER_TOKEN_FILE is used in case —tpc-token-file is not specified

Header Injection (--header or -H)

```
$ xrdcp -f --header "Name: Value" --header "OtherName: Value" http://localhost//remotefile localfile
```

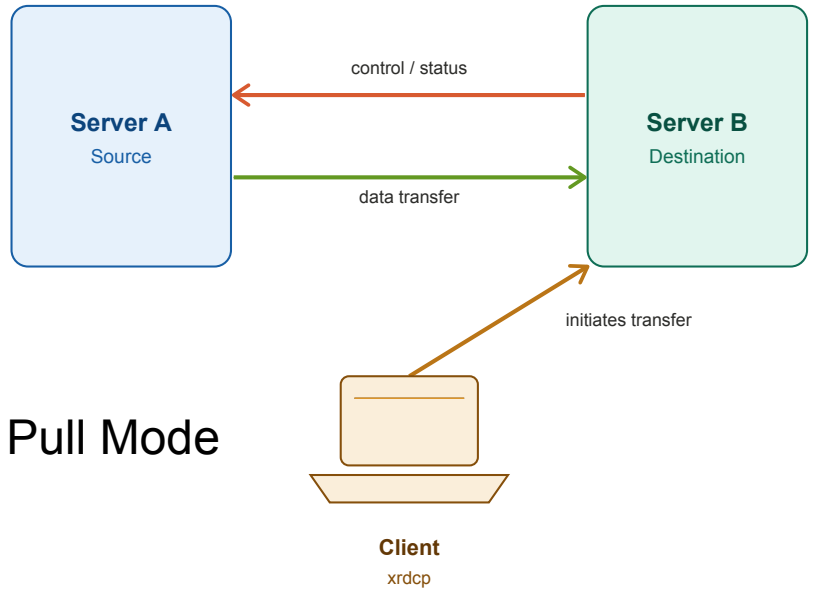
Forbidden headers:

Connection, Content-Length, Expect, Host, Keep-alive, Proxy-authenticate, Proxy-authorization, Proxy-connection, Te, Trailer, Transfer-encoding, Upgrade

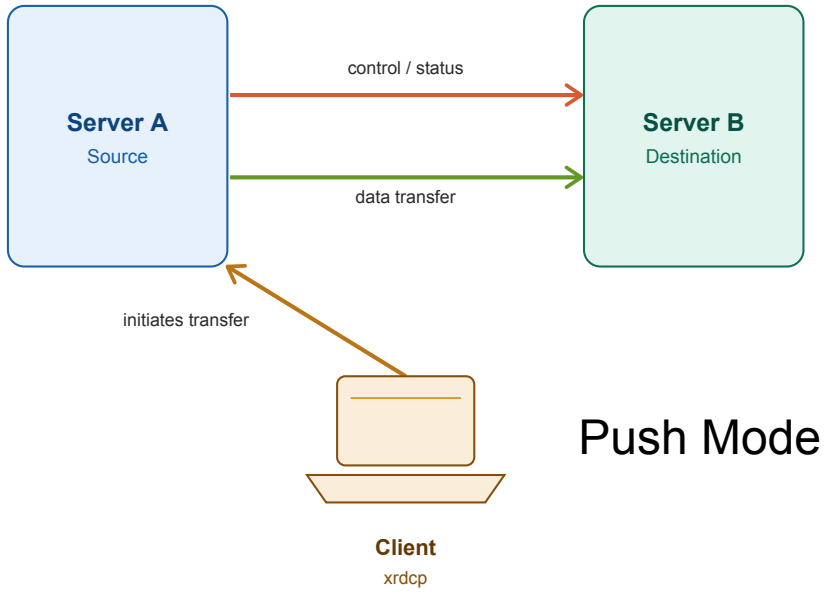
Manpage has the full list!

Header Injection (--header TPC)

```
$ xrdcp -f --header "Name: Value" --header "TransferHeaderName: Value2" -T only \
    http://localhost//sourcefile http://localhost//destinationfile

$ xrdcp -f --header "Name: Value" --header "TransferHeaderName: Value2" -T push only \
    http://localhost//sourcefile http://localhost//destinationfile
```



Pull Mode



Push Mode

Thank you!
Questions?

home.cern