Table of Contents

HTTP-TPC (COPY) protocol updates

Original HTTP COPY specification that was initially implemented may not meet all our future requirements and this document should guide people who would like to propose improvements for the existing HTTP-TPC standard. This page should be used to collect information about all proposals, links to the related meetings or presentations and final decision if / when to implement new extension. Agreement to extend HTTP COPY involve storage developers, transfer tool providers (FTS, gfal2) and experiments / communities who would like to benefit from newly added functionality. WLCG DOMA BDT meetings (indico) or TPC mailing list (`wlcg-doma-tpc AT cern.ch`) should be used to discuss proposals. An update of the HTTP-TPC protocol draft is necessary for all extensions that affects this protocol.

Protocol versioning

Not yet defined, all extensions must be backward compatible with original HTTP COPY specification

Proposed extensions

#	Report Date	Status	Proposer	Short description	Affected components			
					protocol	active	passive	client
1	2022-Aug-23	open	fts-devel	Pass client (FTS) identification to the passive party (DMC-1337)	✗	✗	✗	✓
2	2022-Sep-06	open	RNTWG	SCITAGS HTTP headers (specification)	✗	✓	+	✓
3	2022-Sep-21	open	fts-devel	FTS IPv6 monitoring - perf marker on close (details)	✓	✓	✗	✗
4	2022-Nov-9	accepted	P. Vokac	Monitoring - transfer source and destination addresses (related)	✓	✓	✗	✓
5	2023-May-25	discussion	RNTWG	Include details about TCP re-transmits in performance markers (discussed)	✓	✓	✗	✓
6	????-???-??	implemented	TAPE sites	Tape grouping hints for optimized data recalls	✓	✓	✗	✓
7	2023-June-6	discussion	dCache	Redesign performance markers from scratch (related)	✓	✓	✗	✓

- Decide about features available in FTS/gfal2/SE for GridFTP protocol that are not generally implemented for HTTP (multistream, TCP buffers, timeout for stalled transfers, ...)

List of `TransferHeader` in use

Header	Status	Short description
TransferHeaderAuthorization	standard	Used to authorize active party HTTP request by passive party
TransferHeaderVia	proposed	see: extension #1
TransferHeaderFlowExperiment	proposed	see: extension #2
TransferHeaderFlowActivity	proposed	see: extension #2

Performance markers

!PerfMarker	Type	Status	Short description

!PerfMarker	Type	Status	Short description
Perf Marker ... End		standard mandatory	Performance marker boundary (based on GFD.20)
Timestamp	unix time	standard mandatory	Unix timestamp when active party generated performance marker
Stripe Index	int	standard mandatory	the stripe index for this marker (range of 0 to n where n is the number of stripes)
Stripe Bytes Transferred	bytes	standard mandatory	How many bytes have been transferred by this stripe
Total Stripe Count	int	standard mandatory	The total number of stripes (network endpoint pairs) participating in this transfer
RemoteConnections	list	standard optional	Comma separated network connections tcp:addr:port currently associated with transfer
State	int	dCache proprietary	A machine-readable description of the current status
State description	string	dCache proprietary	A human-readable description of the current status
Stripe Start Time	unix time	dCache proprietary	When the transfer was started
Stripe Last Transferred	unix time	dCache proprietary	When data was last send or received
Stripe Transfer Time	seconds	dCache proprietary	How long the transfer has been running
Stripe Status	enum	dCache proprietary	Current status of the transfer
Stripe Source	proto:addr:port	extension #4 optional	Transfer source address for specific connection
Stripe Destination	proto:addr:port	extension #4 optional	Transfer destination address for specific connection

Discussion / details about proposed extensions

#2: SCITAGS HTTP headers

Description: Add support for scitags (scitags.org) flow identifiers to the HTTP protocol, headers will be generated by transfer client DMC-1344 and consumed by storage which can use them for packet marking (e.g. UDP firefly). Description of SciTags specification include details about HTTP-TPC headers used to pass flow information.

Accepted/rejected: ???? (date + link to meeting or details)

HTTP-TPC standard update pull request: ????

Storage developers plans / releases supporting this feature:

- dCache
- StoRM
- XRootD - xrootd#1984, xrootd#2024
- FTS/gfal2 - DMC-1344
- Rucio - issue#5856

Discussion / meetings

- fts-devel & eos-admin mailing list (June 2023): Scitags HTTP/TPC proposal
- WLCG DOMA BDT (June 7, 2023)
- LHCONE R&D and RNT-WG call (July 5, 2024)

#3: FTS IPv6 monitoring - perf marker on close

Description: although RemoteConnections is optional field in the PerfMarker existing implementations should guarantee it is available on file close. Transferring small files (or not so small over fast networks) doesn't provide performance markers with transfer progress details, because some implementations shows first one only after 5s.

We may decide not to use RemoteConnections in the future, because #4 comes with improved transfer address monitoring.

Accepted/rejected: ???? (date + link to meeting or details)

HTTP-TPC standard update pull request: ????

Storage developers plans / releases supporting this feature:

- dCache
- StoRM
- XRootD
- FTS/gfal2

#4: Monitoring - transfer source and destination addresses

Description: active party in majority of our storage implementations first redirects TPC client to the disknode and only later HTTP-TPC transfer starts, but with dCache real IP address of active party is hidden from TPC client (FTS/gfal2), because headnode internally ask one of available disknode to execute HTTP-TPC transfer. For monitoring purposes (understanding problems with individual disknodes from FTS or central transfer monitoring) it would be useful to have final addresses used during data transfer in the PerfMarker. ~~We need new optional PerfMarker called~~ ~~`Stripe Source`~~ ~~and~~ ~~`Stripe Destination`~~ ~~with source and destination addresses including port number for related connection. The data format for the transfer source and destination follows the same conventions~~ ~~`protocol:address:port`~~ ~~as~~ ~~`RemoteConnections`~~ ~~PerfMarker except in this case it is just one tuple and not a list, e.g.~~

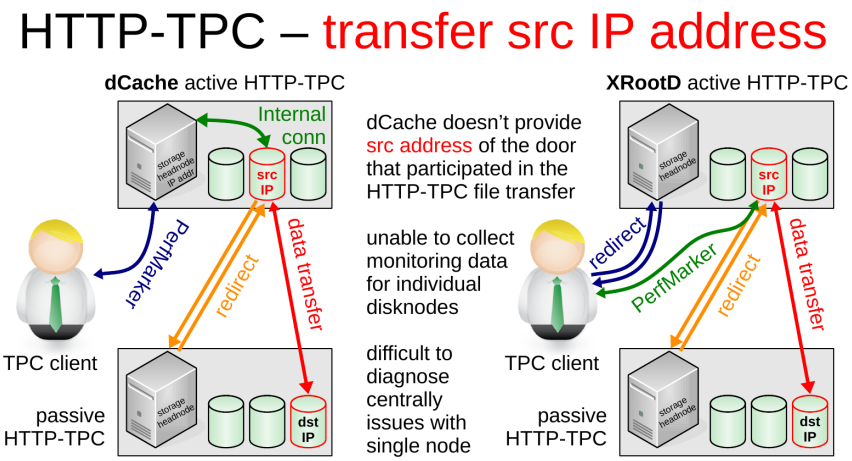
```
Perf Marker\n
Timestamp: 1537788010\n
Stripe Index: 0\n
Stripe Bytes Transferred: 238745\n
Total Stripe Count: 1\n
RemoteConnections: tcp:147.231.25.166:21234,tcp:[2001:718:401:6017:2::28]:24081\n
Stripe Source: tcp:[2001:718:401:6017:2::28]:24081\n
Stripe Destination: tcp:[2001:1458:301:105::100:5]:8443\n
End\n
```

~~Implementation can choose to sent~~ ~~`Stripe Source`~~ ~~and~~ ~~`Stripe Destination`~~ ~~only in the one performance marker for given~~ ~~`Stripe Index.`~~

For XRootD implementation it is too complicated to provide details about individual connections, it is easier to provide just list of all connections, e.g. for transfer done with 2 connections

```
Perf Marker\n
Timestamp: 1537788010\n
Stripe Index: 0\n
Stripe Bytes Transferred: 238745\n
Total Stripe Count: 1\n
RemoteConnections: tcp:147.231.25.166:21234,tcp:[2001:718:401:6017:2::28]:24081\n
Connection: tcp:147.231.25.166:21234:128.142.49.200:8443\n
Connection: tcp:[2001:718:401:6017:2::28]:24082:[2001:1458:301:105::100:5]:8443\n
End\n
```

Clients should prefer these new headers over the original `RemoteConnections`, but it will take several years till this gets in stable storage releases and all sites deploy version with this improvement.



Accepted: February 15, 2023, WLCG DOMA BDT meeting, the goal is to have implementation available by the end of 2023.

HTTP-TPC standard update pull request: ????

Storage developers plans / releases supporting this feature:

- dCache - issue#7058 (already implemented local endpoint info in billing kafka messages)
- StoRM - STOR-1575
- XRootD - issue#1963
- gfal/davix
- FTS

Discussion / meetings

- WLCG DOMA BDT (November 30, 2022)
- WLCG DOMA BDT (February 15, 2023)
- WLCG DOMA BDT (June 7, 2023)

#5: Include details about TCP re-transmits in performance markers

Description: discussed RNTWG Packet Pacing WG that information about TPC re-transmits could be useful for FTS optimizer and as a consequence try to limit packet bursts on the network.

Accepted/rejected: ???? (date + link to meeting or details)

HTTP-TPC standard update pull request: ????

Storage developers plans / releases supporting this feature:

- dCache
- StoRM
- XRootD

Discussion / meetings

- WLCG DOMA BDT (July 19, 2023)

#6: Tape grouping hints for optimized data recalls

Description: New HTTP header `TransferMetadata` used to pass tape archival hints that can be used by site to optimize data recall by placing stored files in sequence that is expected during recalls.

Accepted/rejected:

HTTP-TPC standard update pull request:

Developers plans / releases supporting this feature:

- Rucio
- FTS - FTS-1889 (updates: FTS-1973)
- gfal2 - DMC-1367

Discussion / meetings

- Status of Enabling Archive Metadata for Tapes (June 13, 2023)
- ATLAS Tape grouping hints discussion (October 24, 2023)

#7: Redesign performance markers from scratch

We would like to have details about each HTTP-TPC transfer, but current implementation doesn't really have time to sent performance markers for short transfers. As suggested in the dCache#7441 we need cleaner solution and may be redesign what is sent during HTTP-TPC to the client.

-- PetrVokac - 2022-10-19
