Third-party transfers in WLCG using HTTP
HTTP-TPC: A protocol for moving bulk data using HTTP

We currently have an *opportunity and a need* to migrate the community’s data movement protocols given where GridFTP is in its lifecycle.

- For several years, there’s been ongoing work to develop HTTP to meet our needs.
  - Our small HEP community can leverage the global effort to make HTTPS performant, interoperable, and ubiquitous.
- This builds on a common interpretation of the WebDAV standards, evolving into **HTTP-TPC**.

We have used the last 12 months to greatly mature the implementations and integration with the storage software used in HEP.

- All major storage implementations have demonstrated a HTTP-TPC implementation.
- Except EOS, all have a production version of the protocol.
HTTP-TPC: The Basic Idea

• The client selects either the source or destination to be the **active** side.

• A COPY request is sent, including the headers & URL to use for the real transfer.

• The active side does GET or PUT as needed to move the resource.

---

**Request 1:**
COPY /store/path HTTP/1.1
Host: storage.site1.com
Source: https://storage.site2.com/store/path.src
Authorization: Bearer abcd
Copy-Header: Authorization: Bearer 12345

---

**Third-Party-Copy Client**

---

**Request 2:**
GET /store/path.src HTTP/1.1
Host: storage.site2.com
Authorization: Bearer 12345

---

**storage.site1.com**

---

**storage.site2.com**
Opportunities in HTTP-TPC

• HTTP-TPC is, in the end, just HTTP. You can drive transfers with “any old HTTP client”.
  • curl is enough to move files (but FTS is better!).
  • In fact, our verification tests are written in bash+curl for extra simplicity!
• Flexibility in authentication mechanisms – we’ve demonstrated pure X509, hybrid X509/token, and pure token.
  • The two servers need not have a common mechanism or trust each other; only the client needs to manage the token.
• Only the active side needs to support HTTP-TPC; the other side sees pure HTTP.
  • Example: HTTP-TPC can be used to move directly to/from a Ceph S3 instance.
• Actual data movement can be done using a second protocol.
  • dCache implements HTTP-TPC where data is moved via GridFTP.
• Active server can support multiple streams per transfer, reuse TCP streams across transfers, and load-balance multiple user’s transfers over the same stream.
HTTP-TPC in WLCG DOMA

The WLCG DOMA TPC group aims to cultivate alternatives to GridFTP.

The group runs continuous, nightly testing of 45 endpoints; on a good day, about 40 function.

- No current version of HTTP-TPC implementations require non-standard workarounds.
- We keep a “score” of how many daily tests have succeeded over the past 20 days.
- For functioning endpoints, we then run a low-level test matrix using Rucio.
Isn’t HTTPS slow?

• Perhaps the most common question we get!
• For nearly a decade, TLS encryption has been performed in hardware: yesterday’s server can encrypt faster than today’s network card can send. **Encryption is not a bottleneck!**
• A HTTP host with many transfers in flight should achieve within 10% of its **iperf** speed.
• Most common bottleneck: **TCP**. As with GridFTP, we scale aggregate rates through multiple streams.
  • XRootD implements multi-streamed HTTPS for single transfers: not clear this is worthwhile.

See Poster “Testing the limits of HTTPS single point third party copy transfer over the WAN” for more information!
Authentication and Authorization

All implementations can move file transfers with X509.

- The active side must be able to authenticate with the inactive. This can be accomplished by:
  - **Delegating** the proxy to the active side, OR
  - Having the client use its proxy to **generate a bearer token (preferred)** at the inactive side; the client subsequently passes the bearer token to the active side.
- Bearer tokens provide an enormous amount of flexibility:
  - It’s the defacto authentication mechanism on the Internet, used by other authorization frameworks such as OAuth2.
  - Both sides need not support the same token format.
- Importantly, WLCG has settled on a token profile for VO-issued tokens: this provides the path forward for a **interoperable, “X509-free” authorization**.
Authorization

The WLCG JWT profile finalized last month – with:

• SciTokens client library has committed to being “dual-profile”; existing users.
• As with SciTokens, this is a very lightweight layer on top of JWT. dCache, XRootD, and StoRM are racing to finish support.

Basic idea: allow VOs to issue tokens that dictate the file access permissions inside their own storage areas.

Next up? Using token exchange to allow FTS to start transfers without proxies.

• Currently in the design phase…

For more information, see Track 3 “WLCG Authorization; from X.509 to Tokens” later today and “Beyond X.509: authN and authZ in practice” on Thursday
For several years, the XRootD server software has been multi-protocol: instances can speak either the xrootd or HTTPS protocol. Recent improvements:

- Implemented RFCs for checksum calculation.
- Add support for OAuth2-based token request. Internal token format is based on macaroons.
- Matured implementation via a steady stream of bugfixes: no HTTP-TPC work-arounds needed for clients!

To enable HTTP in your config:

```
xrd.protocol http:1094 libXrdHttp.so
http.cadir /etc/grid-security/certificates
http.cert /etc/grid-security/xrd/xrdcert.pem
http.key /etc/grid-security/xrd/xrdkey.pem
http.secxtractor /usr/lib64/libXrdLcmaps.so
http.listingdeny yes
http.staticpreload http://static/robots.txt
    /etc/xrootd/robots.txt
http.desthttps yes
```
Implementation progress over the last year - dCache

dCache support for macaroon-based HTTP-TPC is feature complete for well over a year, so not much development.

- In DOMA testing, dCache has the most test endpoints and provides very reliable, consistent test results.

HTTP-TPC works out-of-the-box with default configuration:

- However, it’s possible to configure dCache so it doesn’t work,
- Some sites need to update their configuration.

Note this is separate from ATLAS’ move away from SRM

- Changes also needed to support non-SRM based uploads.

Future work:

- Help sites enable HTTP-TPC in production instances,
- Update dCache’s SciToken support to allow SciToken-authorised macaroon requests,
- Add support for WLCG AuthZ JWT tokens.
- Participate in HTTP-Token testbed,
- Additional monitoring.
Implementation progress over the last year - StoRM

WebDAV third-party transfer introduced in v. 1.1.0
• In production!

Token-based authorization and delegation
• No GSI/Gridsite delegation support
• OAuth2 endpoint for VOMS credential → token translation

Support for external trusted OAuth2/OIDC authorization server
• Coarse-grained, VO-level authZ

Current work:
• WLCG JWT profile support
• Fine-grained capability/group-based authZ

See poster for more info
There is a healthy community building around HTTP-TPC!

What’s next for HTTP-TPC?

• Work with the WLCG DOMA TPC group to continue rolling out support at additional sites.
  • We are still missing a handful of T1 sites as well as a CERN EOS instance.
• Absolutely need to get EOS instances deployed! Last major storage system that hasn’t deployed an endpoint for testing.
• Tackle completely X509-free file movement with FTS.
• Work with the experiments to move production data through their transfer frameworks.
  • Goal: US CMS would like to have at least one site with 30% of its traffic through HTTP-TPC.
We need **YOU** in the HTTP-TPC Transfer Tests!

Questions?