WLCG DOMA TPC Working Group

Brian Bockelman
USCMS Tier-2 Meeting

Jargon Decoder:
DOMA = Data Organization, Management, and Access
TPC = Third Party Copy (i.e., what GridFTP does!)
Third-Party-Copy (TPC)

- Third party copy - moving data from site A to site B - is one of the core, essential activities on the grid.

- Many individuals have been working on improving approaches and techniques continuously!

- It is suddenly more interesting because, as a community, we have realized a need to replace the functionality found in the Globus Toolkit (particularly, GridFTP & GSI).

- Globus is not the sole reason but rather the catalyst.
We formed the working group shortly after CHEP.

- Aim is to **put in place viable alternate protocol(s) for GridFTP**.
  - Participants are currently working on both XRootD and HTTP/WebDAV.

We break down the work to three phases:

1. **Prototype / implementation**: Demonstrate viability of protocols. Ensure all storage implementations have a valid alternate in production.

2. **Early deployment**: Ensure rollout of alternates at all sites with >3PB storage.

3. **Widespread deployment**: Rollout to remaining WLCG sites.
Site Participants
Production sites, testbeds, developer instances

- RAL
- Prague
- Glasgow
- Brunel
- Manchester
- Lancaster
- CERN DPM trunk
- CERN DPM release candidate
- DESY developer testbed
- DESY dedicated testbed
- AGLT2
- BNL
- Imperial College
- PIC
- CERN EOS pre-production
- INFN-T1
- Queen Mary University
- NERSC
- SLAC
- OU
- Nebraska
- Bonn
- FNAL
- Beijing
- DynaCloud CERN
- DynaCloud CERN (Grid instance)
- IN2P3
- Brussels
- Florida
- SURFSara (production & test)
- NDGF
- University of Victoria
- TRIUMF
Basic Idea

- Both protocols convert the third-party-copy to a normal download that is performed by an “active” side.

- HTTP can do either push or pull: as only one side needs to understand COPY, the other side can be a “pure” HTTP server - nginx, Apache, etc.

Note: at least theoretically, the two requests could be done with different protocols!
Tribulations

• In late 2018, we setup a dedicated Rucio instance to drive transfers between sites.

  • Around January 2019, started scale testing HTTP transfers.

  • At this point, regularly moving >500TB / week across participating scale test sites.

• So many sites participating in the test transfers that we are breaking the Kibana plots. Split out over the next few slides!
Transfer Connectivity

Do not pay attention to any given site - who is “red” and who is “green” when the snapshot was taken. This evolves daily.
Focus on the breadth of the testing!
Transfer Connectivity

Do not pay attention to any given site - who is “red” and who is “green” when the snapshot was taken. This evolves daily.
Focus on the breadth of the testing!
Data Rates

Each bar is a 6-hour time window.
Data Transfers

- Both XRootD and HTTPS implementations support load-balancing transfers over multiple TCP streams.

  - Honestly, this appears to provide minimal benefit these days. TCP is much better than it used to be — and, as a community, we tend to have multiple files in flight as opposed to a single 100Gbps transfer.

- We had some internal questions about the cost of encryption for HTTPS. Luckily, almost all the heavy lifting is done by hardware, no longer software.

  - Quick tests show that a modern server can perform TLS encryption at ~385Gbps; serving over HTTPS, one can mostly fill a 40Gbps connection (tests limited by disk I/O).

  - A 7-year-old server can do ~20Gbps of encryption and serve at 10Gbps over HTTPS (tests limited by network card).
Nightly Smoke Tests

• It’s difficult to digest a simple site status from transfer matrices: Is it my site that is broke? Is it the other end?

• Paul Millar has implemented some simple tests for HTTP TPC functionality against a “known good” endpoint.

• These get sent out nightly. It’s driven by a “bash script invoking curl”; any admin should be able to reproduce failures at their site.
Storage Status

Can you do transfers with FTS?

• **XRootD (SLAC)**: Needs at least Xrootd 4.9.1; in latest OSG release.
  
  • CMS-specific instructions available.

• Needs:
  
  • Writes enabled via Xrootd (done in 2018, hopefully!)
  
  • HTTP enabled (done in 2018)
  
  • Enable checksum verification.
  
  • Enable Macaroons.
  
  • Enable third-party-copy plugin (can test via fts3-devel.cern.ch).

• There’s additional PhEDEx changes needed to export HTTPS and a one-line patch
Site Status

• **Caltech**: PhEDEx PFN includes double-slash (arguably a bug in Xrootd; easy to workaround)

• **Wisconsin**: Macaroons not working.

• **UCSD**: Macaroons not working (might need a Xrootd upgrade to 4.9.1?)

• **Purdue**: TFC issues; PFNs of the form davs://xrootd.rcac.purdue.edu:1094//store/PhEDEx_LoadTest07/LoadTest07_Debug_US_Purdue/US_Nebraska/5172/LoadTest07_Purdue_D4_NoOqUBw2Bs5PjgjE_5172

• **Florida**: Appears checksums are not enabled for all servers?

• So, some work to do — but making good progress!
We Want You!

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

• If you aren’t there already, get your name on the list!

• Peruse the above documentation and get HTTP transfers working. Part of the 2019 plans for USCMS anyway!

  • Only additional work is to set aside a separate storage directory (and authorization) for “dteam” VO (dteam = development team; used historically by EGI).

• Add your site info to the table so we can test the endpoint.

• Join the mailing list, introduce yourself. Ask to be added to the functionality test matrix.

• CMS-specific tests ongoing with Florida, Purdue, UCSD, Caltech, Nebraska, and Wisconsin.

• Missing MIT and Vanderbilt.
Questions?

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