

HTTP solutions for data access, transfer, federation

Fabrizio Furano (presenter)
on behalf of the IT-GT/DMS crew

- Intro
 - Features http+dav
 - Our challenge
- DPM
- Federations
- FTS
- Davix

- HTTP: open and extensible can be used in many app domains
 - Standard in the way things can be embedded
 - Supports features that became a standard, like WebDAV, or metalinks
- **Data access, data xfer and federation CAN be done**
 - **We all agree.** At the same time it's difficult to find elsewhere a complete HEP production level storage/data access system that does what seems obvious to us.
- There are quality components that support the low level features (e.g. protocol implementation)
 - The Apache framework is a good example
- Our challenge: build on them quality components that fulfil the HEP requirements
 - Clustering of disk backends into storage pools + Clustering of pools into sites (DPM)
 - Keep the historical technologies available until they are used (SRM, dpns)
 - Provide the needed data access protocols (GridFTP, HTTP, Xrootd, ...)
 - HTTP/DAV support for XROOTD is also being worked on
 - Clustering of sites into federations
 - Static, explicit indexing of content of sites (LFC)
 - Dynamic high performance discovery (Dynamic feds)
 - Sophisticated, scheduled transfers (FTS)
- This comes from EMI, the goal and the effort is shared among DPM, dCache and STORM
 - NEWS: the STORM people now have a prototype for HTTP/DAV

- The idea is to keep it until it's used
- Missing areas of SRM that could be implemented over HTTP
- Get/set checksums
 - Can already get checksums
 - Could extend DAV implementation to write them
- Write to spacetoken is possible
 - `<...>?spacetoken=reservation`
- Query usage would need some implementation via namespace
 - RFC2518 “Quota and size properties for DAV”

- Rationale:
 - Evolve smoothly from the historical design
 - Make integration with other frontends/backends easier
- Goals
 - support the big DPM community
 - accommodate many types of data access frontends
 - evolve frontends to HTTP/DAV with new high quality components
 - accommodate many types of backend (legacy pools, S3, HDFS, etc.)

- Frontend based on Apache2 + mod_dav, using DMLite
- Supports all the historical features, mapped to DAV
 - We implemented the old Cns API on top of DAV, no need to change the applications
- Can be for both get/put style (=GridFTP) or direct access
 - Some extras for full GridFTP equivalence
 - Multiple streams with Range/Content-Range
 - Third party copies using WebDAV COPY + Gridsite Delegation
 - Random I/O
 - Possible to do vector reads and other optimizations
- Metalink support (failover, retrial)
- Since 1.8.4 it is already DMLite based

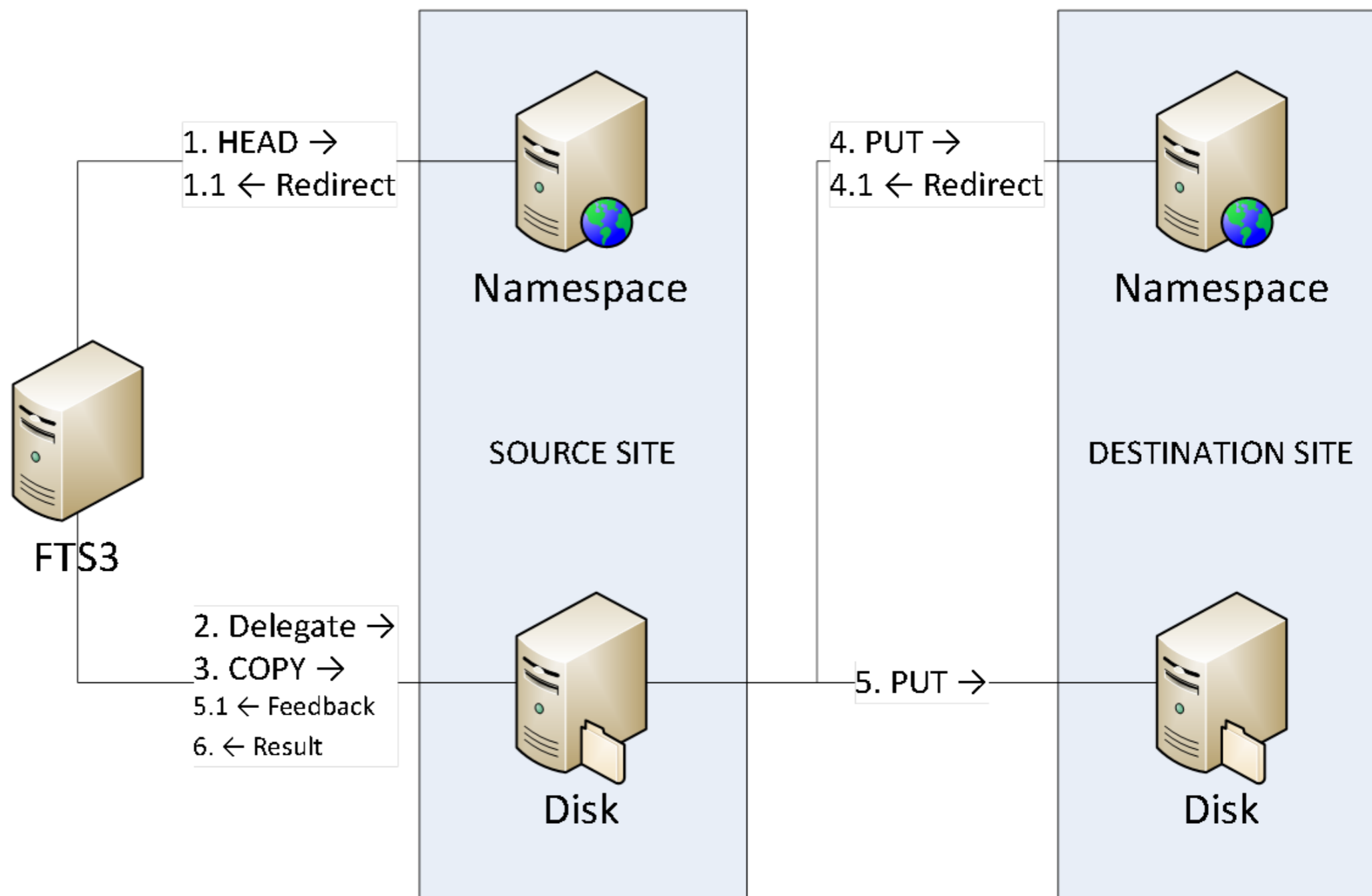
<https://svnweb.cern.ch/trac/lcgdm/wiki/Dpm/WebDAV>

- Together with our partners (in particular ASGC) we run many rounds of performance tests
- The conclusion is that the HTTP data access is becoming competitive for the main use cases (Xfer, analysis, LAN, WAN)
- Many aspects
 - Tuning (Apache+backends in our case)
 - Technology
 - TTreeCache in ROOT is very important for some use cases
- No space here for tables... please ask if you are interested

- One element in the HTTP ecosystem is the potential of doing this WAN loose clustering, that we often call “federations”
- HTTP is a good data access protocol. Any browser supports the three points:
 - Native support for redirections
 - Usable (=Good performance and robustness) over WAN
 - Support for direct data access and metadata access
- HTTP can also be used as a clustering control protocol.
 - We can federate HTTP and DAV servers on the fly

- We've demonstrated that HTTP WAN transfers have the necessary performance
- Enabled 3rd party transfer on DPM (and it's coming in dCache)
 - Currently works in “push mode”, any sink is supported
- We developed a complete client API using HTTP
- We've integrated HTTP support into GFAL2
 - now gfal2 offers a way to integrate HTTP resources easily into grid data management in general
- FTS3's HTTP support is based on gfal2's HTTP support

- Significant effort in the federation domain. The same DMLite + WebDAV stack supports:
 - LFC: can be thought as an HTTP redirector that
 - allows us to browse the full namespace
 - maps logical names into physical names
 - accepts file replica registration requests
 - Dynamic Federations: namespace parts can also be reconstructed quickly on the fly
 - With no need of registering replicas, they are where they can be found (and their locations cached)
 - High resiliency to sites/clusters up-downtimes
 - High resiliency to network glitches
 - Great flexibility, same access/browsing interface as LFC



- For clusterization and HEP access to work we need some basic bricks:
 - Redirection support in all the primitives of HTTP and DAV
 - Flexible support for the various security protocols (e.g. GSI proxies, krb, pwd, x509)
 - Map the “posix-like” things (open, stat, close, ...) into DAV requests/responses
- **Data access, data xfer and federation CAN be done with standard clients**
 - **We all agree.** At the same time it's difficult to find elsewhere a complete HEP production level storage/data client that does what seems obvious to us.
 - The features that the users see is also proportional to the completeness of the clients they use and to their quality.
 - Forcing app developers to implement features over XML parsing is ugly. This is the option without a complete client.
 - We can use a standard client to implement what we need, on top of it.

- Support for all these things had to be encapsulated in a good client
 - Maybe this had to be a contribution from HEP to the HTTP world
 - Feature support in the various clients is sparse
 - We can use the standard clients as low-level clients, like libNEON, and add the features
- HTTP/WebDAV is a protocol, Data access is an application.
 - Users want quality tools, not a specification
 - Current HTTP tools are designed around generic HTTP requests
 - Our use case is high performance file access and file management
- Our DAVIX client could be integrated in ROOT, modulo the TWebFile/TWebSystem wrapper
 - This part is under discussion and investigation

GT

Thank you

... questions?

- LCGDM Wiki:
 - <https://svnweb.cern.ch/trac/lcgdm>
 - Contains links to tech documents, papers, presentations, information, etc.
- Some papers:
 - DPM: <https://cdsweb.cern.ch/record/1458022>
 - Dyn Feds: <https://cdsweb.cern.ch/record/1460525>
- Some people to talk to:
 - DPM, DMLite: Ricardo Rocha (ricardo.rocha@cern.ch)
 - Dynamic Federations: Fabrizio Furano (fabrizio.furano@cern.ch)
 - DAV, DMLite: Alejandro Alvarez (alejandro.alvarez.ayllon@cern.ch)
 - DAVIX, GFAL: Adrien Devresse (adrien.devresse@cern.ch)
 - FTS3: Michail Salichos (michail.salichos@cern.ch)