Dynafed as a DTN/TPC agent

Sept 2018

Fabrizio Furano, Oliver Keeble, Andrea Manzi

Dynafed and TPC

- Dynafed does “storage federations”, used so far only for HTTP
  - Constantly monitors a set of endpoints
  - Makes redirection choices for file GET/PUT requests
  - Effectively fakes the existence of a friendly namespace to browse

- It uses the same Apache frontend used by DPM, which does have the third party copy (normally disabled)
- At some point we were reasoning… what if we enable it? How much effort will it need?
- This entered in a more generic round of refurbishment of the DMLite API, adding TPC calls to it

TPC Features in Dynafed

- The core implementation is not yet polished, yet it works sufficiently well to give a preview. The features are pretty original

- The “redirector” of an HTTP federation accepts COPY requests
  - Can redirect them to an endpoint known to support them
  - Can tunnel the data if no suitable endpoint is able to process COPY requests. In this case **it can silently translate the protocol on the fly**, e.g. a 3cp http->gridftp or xrootd->http or others

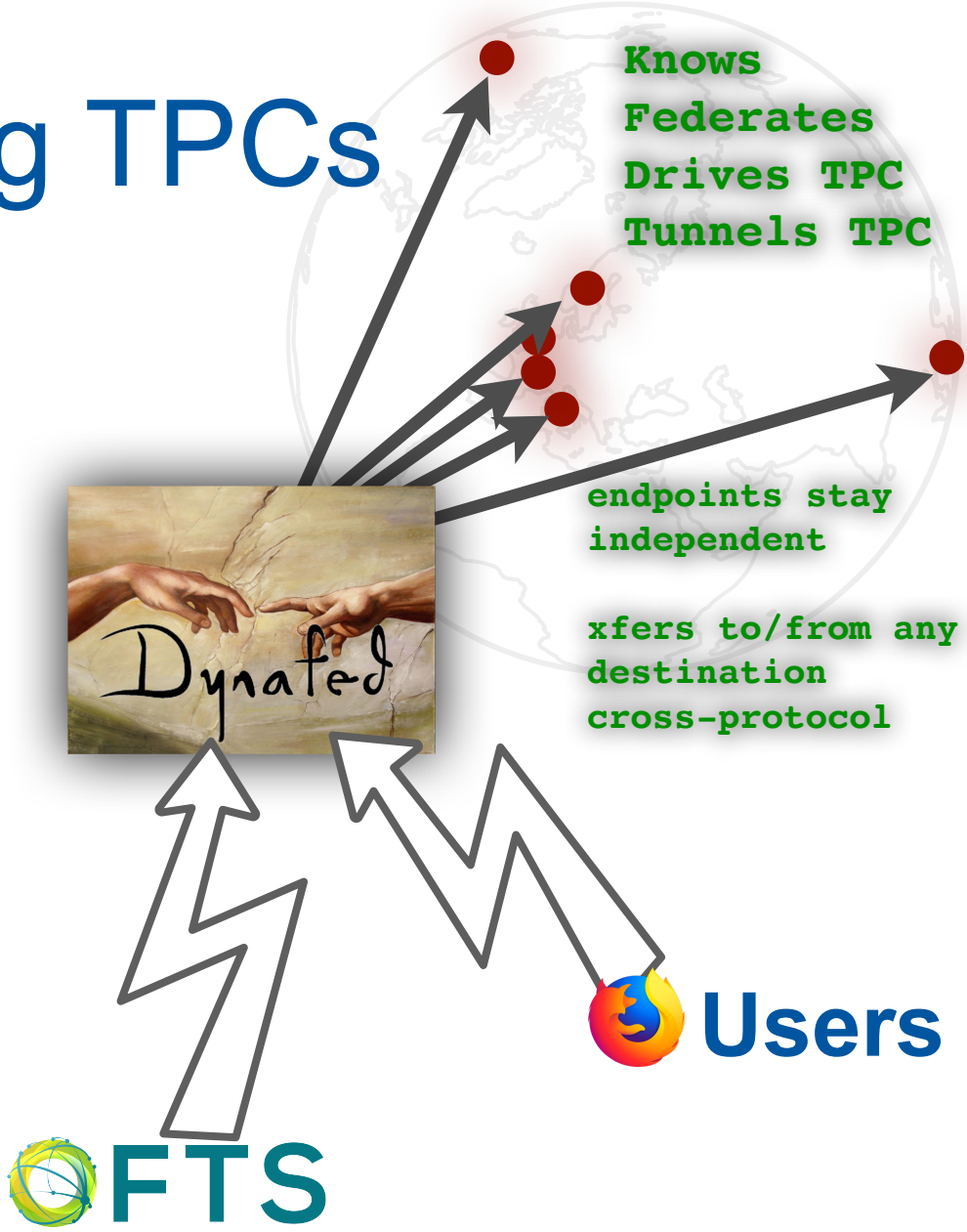
- **An HTTP federation becomes able to fulfil COPY requests**, independently from the mix of endpoint types that it contains, e.g. DPM, dCache, AWS S3
- **An HTTP federation can work as a scalable file transfer agent**
- The interesting part is that it’s browseable, Dynafed style, and that it knows in realtime the upness of its known, federated endpoints

- Someone commented that this is a Data Transfer Node (DTN). Surely a flavour of it

Dynafed managing TPCs

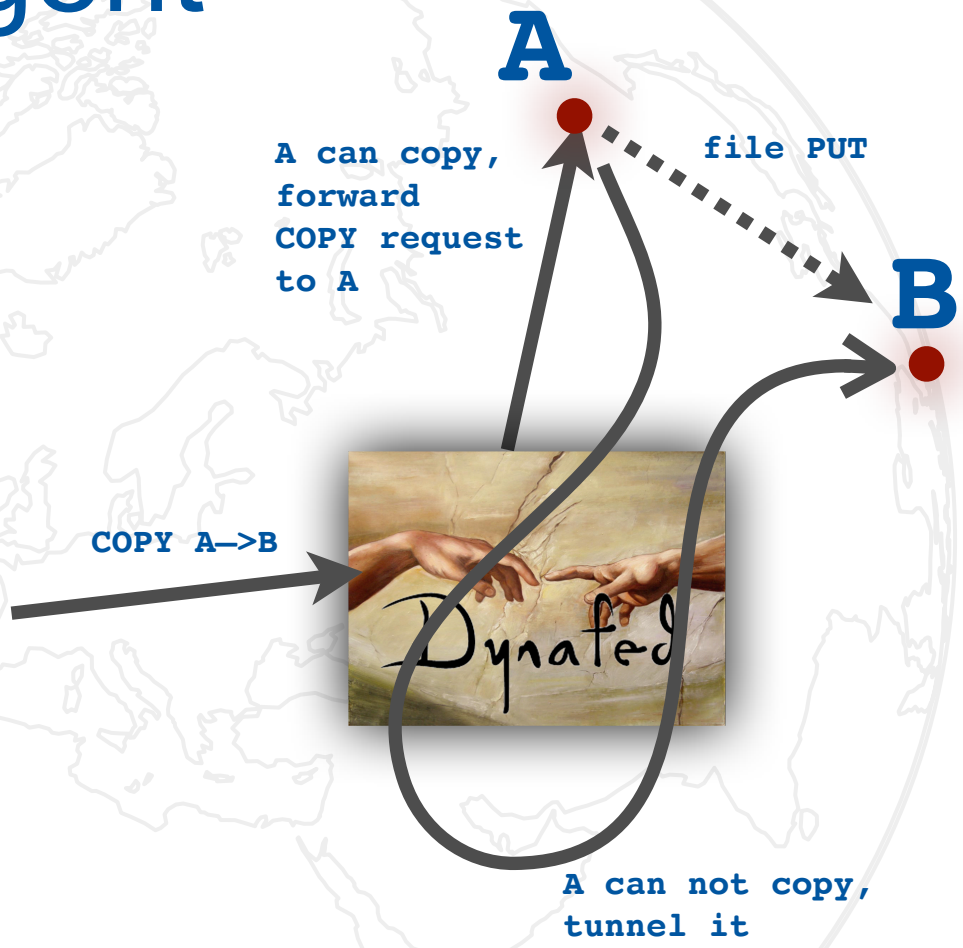
- "DTN" style operation, manages TPCs
  - Can forward a TPC request to a capable host or tunnel it
  - Enable data movement for non-TPC storage (e.g. cloud)
  - Enable cross-protocol data movements

- A scalable geographical agent that manages 3rd-party copy tasks on behalf of authorised requestors
- Can work globally, regionally, individually
- All the federation-related features (e.g. browsing, locating) are untouched



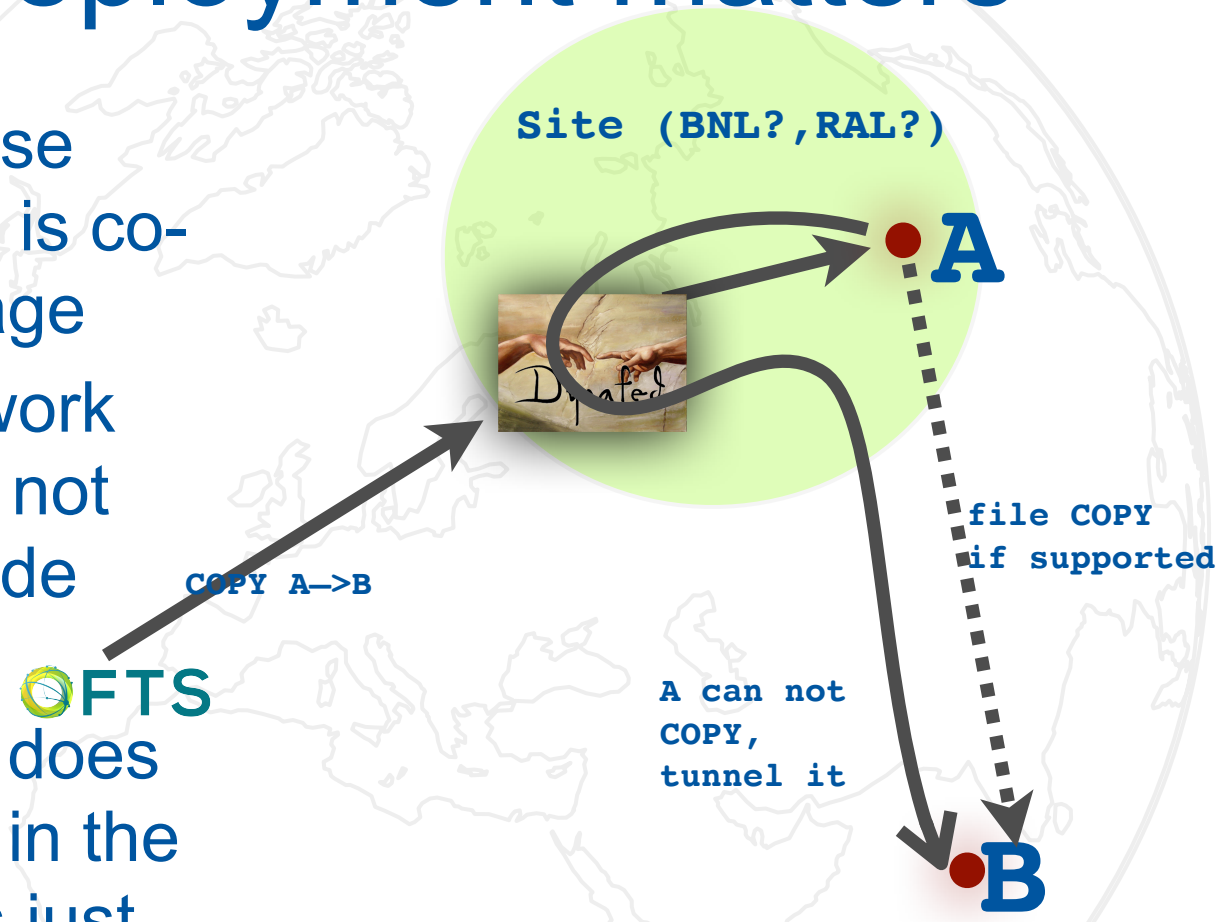
A smart transfer agent

- There are many combinations, including the cross-protocol ones
  - e.g. COPYing from gsiftp to root or HTTP, and others
- Here's a simple one. Dynafed federates A
- A client (e.g. FTS) asks Dynafed to COPY a file from site A to site B
- If site A is able to do it, then the COPY request is forwarded to A
  - (the COPY performance markers are then forwarded backwards, from A to the client)
- If site A can't do it (e.g. because it's an S3 bucket) then Dynafed will tunnel the COPY
  - (and send the performance markers back to the client)



Colocation - deployment matters

- An interesting use case arises when Dynafed is co-located with the storage
- COPY requests will work even if the storage is not accessible from outside
- Funnily enough, the "datamover process" does not even need to run in the dynafed machine. It's just a little script.



Dynafed, lcgdm-dav and DPM

- Dynafed shares the frontend Apache modules with DPM
- The bulk of this work is exactly there, activated by an Apache flag (normally off, for the DPM normal behaviour)
- This flag makes mod_lcgdm_dav simply forward the internal TPC calls to the dmlite layer, where dmlite plugins can give their implementation
  - Instead by now TPCs are implemented privately by the various frontends, xrootd, apache, gridftp
- **We just saw a preview of a simple Dynafed implementation**
- **In the future, we may think at doing a similar thing for DPM, and have just one mechanism that manages TPC for all protocols (except gridftp I believe)**
  - One mechanism means that the DOME daemon in DPM may properly queue and schedule TPCs, like it does for checksums
    - That would mean preventing TPC overload, not just relying on FTS to heuristically detect it
  - DOME already has all the low level components to do this. This is tempting and relatively cheap to implement. Will see next year.

Work in progress

- Verify all the main combinations, most of them work
- Transferring the “FTS/GFAL performance markers works
- Verify that bearer tokens are properly handled
- Improve the error reporting

- A few minor issues in gfal-copy
  - the envvar BEARER_TOKEN breaks S3 presigned URLs
  - need to add support in gfal-copy for multiple tokens (hopefully not through envvar hell)

- Testing, testing, did I say it? Testing!