CERN

Michal Simon

XRootD workshop report

Outline

- XRootD status and plans
- Xcache
- Delegation & TPC
- Experiments
- HTTP ecosystem

XRootD workshop overview

- France, Lyon, IN2P3 (11-12/06/2019)
- ~20 registered participants (~5 remote)
- 1.5 days presentations from users and the XRootD team

XRootD status and plans

- Current adoption 4.9.x (4.9.1 recommended)
  - Though many still run 4.8.x (or even older)

- The upcoming feature release is 4.10.0
  - RC5 cut last Friday
  - To be released tomorrow

- On the horizon R5 (Q3 2019)

XRootD status: 4.9.x highlights

- Client: redirections on stateful operations, **redirect trace-back**, force disconnect, ZIP support bug fixes, declarative API
- Server: enhancements for containerization, XCache ingest via HTTP, **Macaroon support**, **SAN support**
- Client/Server: **TPC-lite with delegation**, vector writes

XRootD status: 4.10.x highlights

- Client: retry/redirect policy changes for better metamanager/metalink utilization, **chunked response** (dirls)
- Server: multiple XCache write back streams, direct XCache access, additional clustering options for XCache
- Client/Server: new **prepare plug-in and interface**

XRootD status: R5 highlights

- Client: channel level plug-ins, review ABI and public headers
- Server: sec entity refactoring, monitoring g-stream (periodic medium level information, JSON/XML), trivialize OFS plug-in wrapping
- Client/Server: extended attributes, **encryption**, extended stat

XRootD status: post R5 highlights

- Client: RDMA support, dynamic data source selection
- Server: recursive delete, uid/gid tracking
- Client/Server: **bundled requests**, **verified close**, **streaming read**

XRootD status and plans

- Server: https://indico.cern.ch/event/727208/contributions/3444600/attachments/1859800/3056084/IN2P3-190611-XRootD_Server_Status_and_Plans.pdf

- Client: https://indico.cern.ch/event/727208/contributions/3444599/attachments/1859886/3056268/xrdwrkshp_status.pdf

XCache status and plans

- Serve data to local clients (origin: remote data source – usually federation)
  - Data are read in blocks, prefetching is optional
  - Store data on disk via write queue
  - Purge old files as disk gets full

- Standard XRootD client used to access remote server (behaves as xrdcp, also in terms of authentication)

XCache status

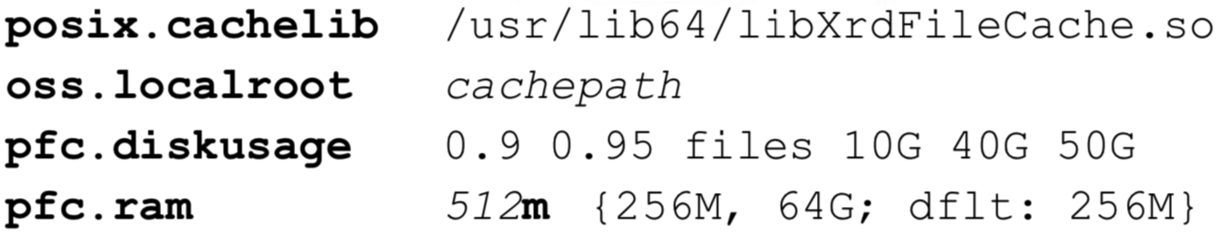
- Minimal Xcache setup / configuration

- List of **features** and **configuration** options:
  - pfc.blocksize (large buffer for whole file streaming vs small for vector reads),
  - pfc.origin vs authenticaction
  - pfc.prefetch, pfc.ram, pfc.writequeue, pfc.diskusage, pfc.spaces, pfc.decision, pfc.trace

XCache status

- Serveless / **client-side caching**
  - Available only through the POSIX interface
  - Single process only

- Minimal config / setup



```
posix.cachelib   /usr/lib64/libXrdFileCache.so
oss.localroot    cachepath
pfc.diskusage    0.9 0.95 files 10G 40G 50G
pfc.ram          512m {256M, 64G; dflt: 256M}
```

XCache plans

- **Monitoring**, **quota based purging**
- Optimizations (RAM management, block selection for prefetching, etc.)
- Efficient running with smaller block sizes
- XCache clusters (balancing disk usage and staging frequency)
- Dealing with **corrupted files**

XCache status and plans

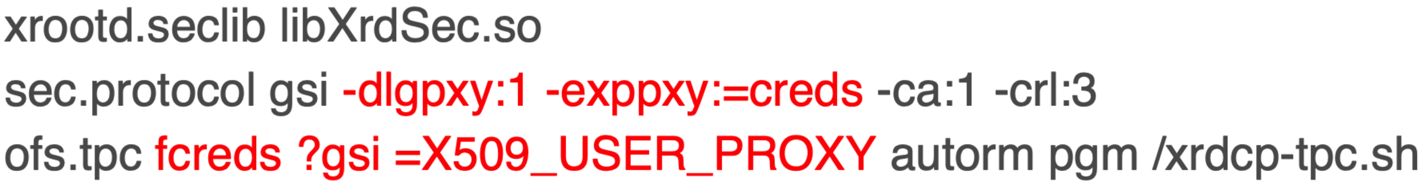
https://indico.cern.ch/event/727208/contributions/3444604/attachments/1859894/3056280/XCache-FeaturesEtc-Lyon-2019.pdf

Delegation and TPC

- GSI authentication (verify server cert using known CAs, RFC 2818, SAN, verify client DN using known CAs)

- **X509 proxy delegation** (server: generate RSA pair, client: sign public key using its X509 proxy + voms attributes)

- Server side configuration:



```
xrootd.seclib libXrdSec.so
sec.protocol gsi -dlgpxy:1 -exppxy:=creds -ca:1 -crl:3
ofs.tpc fcreds ?gsi =X509_USER_PROXY autorm pgm /xrdcp-tpc.sh
```

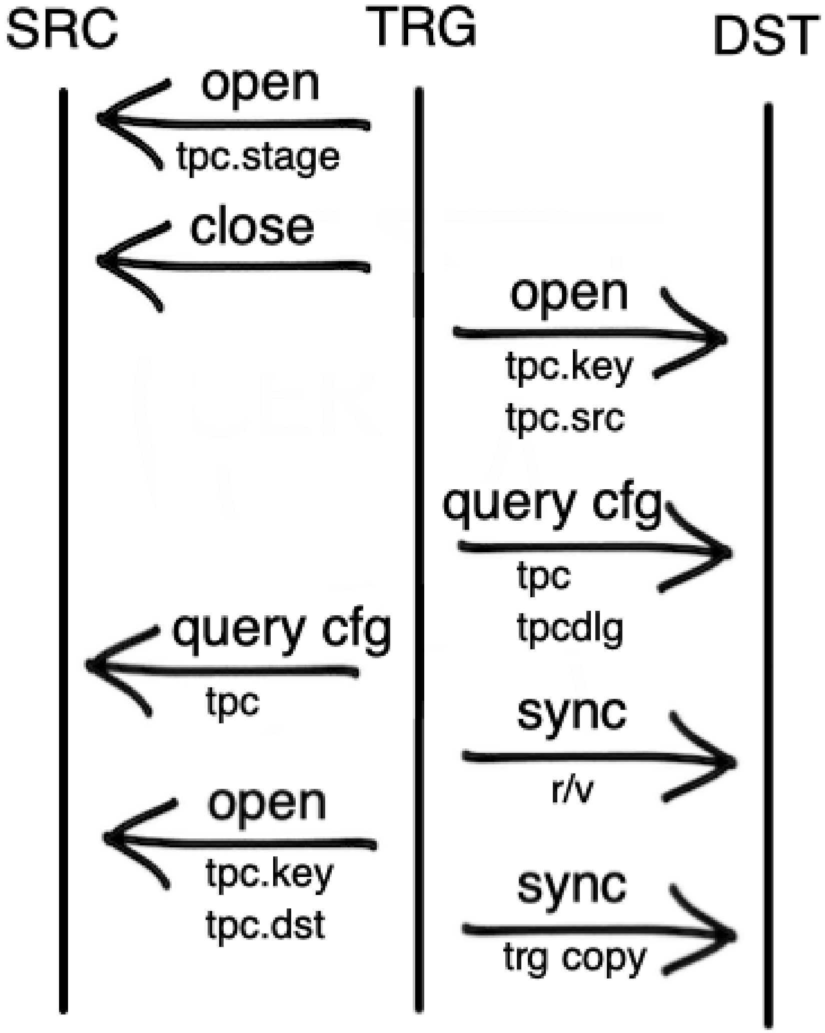
Delegation and TPC

- Extend vanilla TPC to WLCG storage ecosystem (X509 proxy delegation)

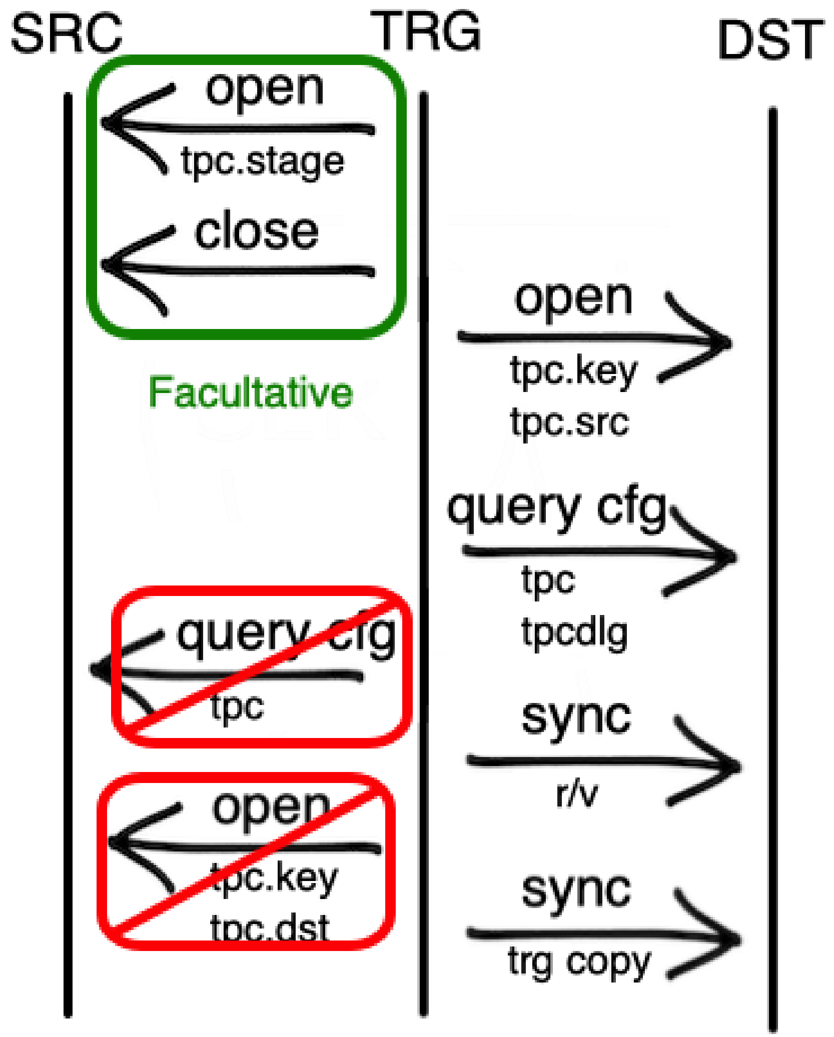
- All storage use redirections to data servers
  - EOS, DPM & dCache add additional AA info in CGI so TPC script must start from head-node

- Destination maybe XRootD proxy
  - May choose to write directly to backend storage

Delegation and TPC



Delegation and TPC



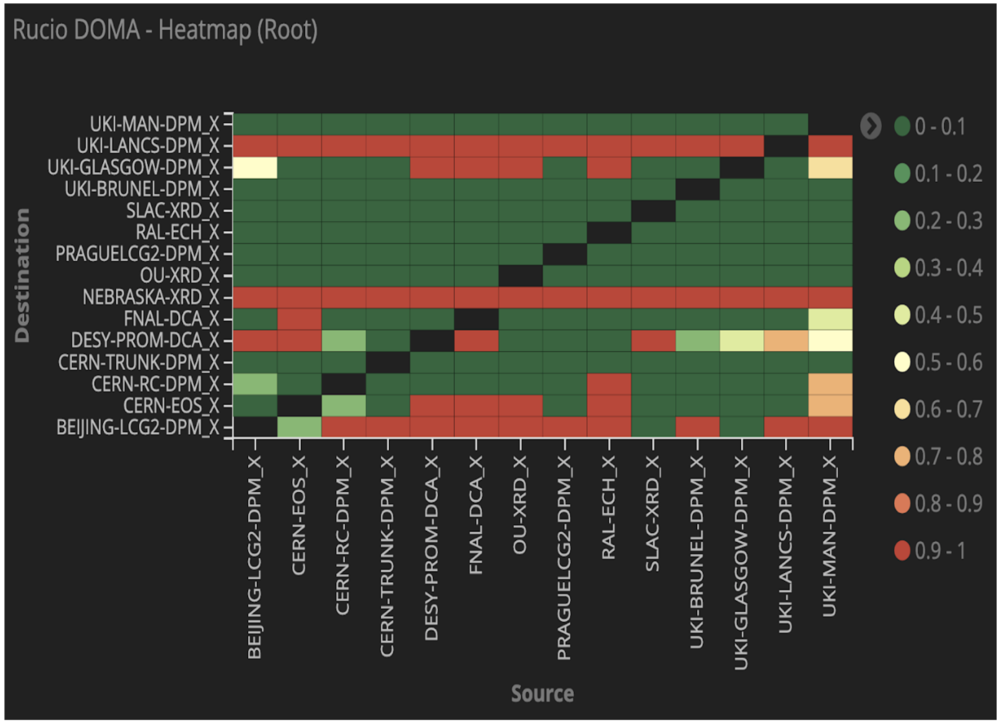
Delegation and TPC: status

- Two implementations: native (XRootD, ECHO, DPM, Storm) and Java for dCache

- EOS is bit different as it has it’s own TPC plug-in (working solution exists)

Delegation and TPC: status

- The coverage phase is completed,
- Entering the phase of early deployment
- FTS works with XRootD TPC & delegation



Delegation and TPC

- Delegation: https://docs.google.com/presentation/d/1fZrFP9U23p9RalIqdA7ivTgiRuKyEQA3kqvIkSS_nmw/edit#slide=id.p

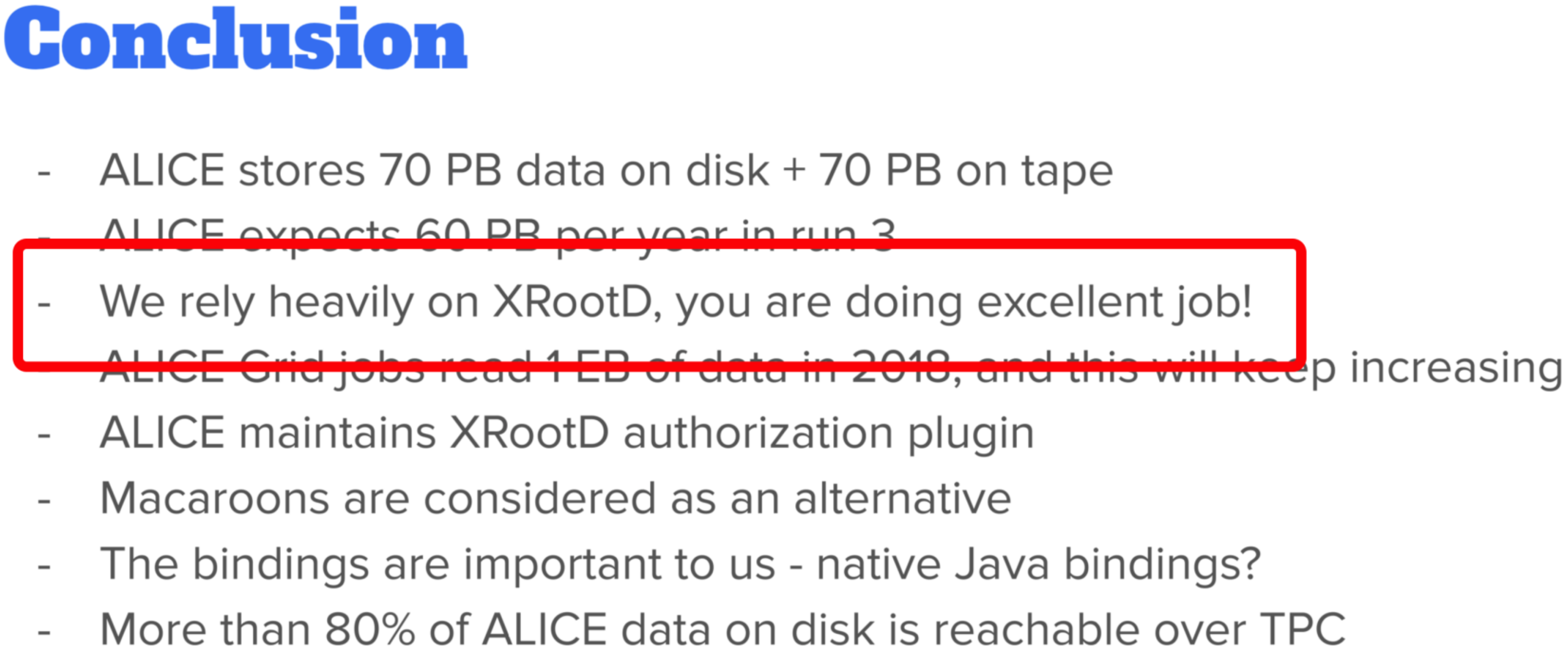
- TPC: https://docs.google.com/presentation/d/1Wt1l5XhUqFN5k27Gc5PmN-2OBEnnveGEu1ky1OzrjC4/edit#slide=id.p

Experiments

- Atlas: centrally managed caches with kubernetes, SLATE and k8s
  - https://docs.google.com/presentation/d/1dijMTIyKULNtuDnEYboEwc-V6ct8SIsc0NxzxKoMppQ/edit#slide=id.p
- CMS: remote file access latency has big impact on CMS analytics jobs, XCache helps with latency hiding
  - https://indico.cern.ch/event/727208/contributions/3444609/attachments/1860136/3056787/XRootD-Workshop-201906.pdf

Experiments

- Alice:



Conclusion

- ALICE stores 70 PB data on disk + 70 PB on tape
- ALICE expects 60 PB per year in run 3
- We rely heavily on XRootD, you are doing excellent job!
- ALICE Grid jobs read 1 EB of data in 2018, and this will keep increasing
- ALICE maintains XRootD authorization plugin
- Macaroons are considered as an alternative
- The bindings are important to us - native Java bindings?
- More than 80% of ALICE data on disk is reachable over TPC

HTTP ecosystem

- XrdHttp: simplicity (compared to Apache), TPC plug-in, Macaroons authorization support, SciTokens authorization support
  - Port sharing with XRootD or vanilla HTTP port (80) through systemd socket inheritance
  - Security extractor plug-ins – recognize VOMS
  - HTTPS -> HTTP + security token (used e.g. in DPM)

- HTTP ingestion to XCache

HTTP ecosystem

- XrdHTTP: https://indico.cern.ch/event/727208/contributions/3444621/attachments/1860746/3058009/XrdHTTP.pdf

- HTTP ingestion to Xcache: https://indico.cern.ch/event/727208/contributions/3444622/attachments/1860256/3057010/Adding_HTTP_Ingestion_Support_to_XCache.pdf

Summary

- Overall positive feedback from users on the XRootD framework
- XRootD TPC with delegation is production ready
- Increasing roll of XCache
- Encryption will be the enabling factor for access token in root/xroot protocol
- Thanks a lot for participation and feedback ☺

Question?