Storage Interface: LHCb viewpoint

Ph.Charpentier, CERN/PH

Some definitions (just in case…)

- LFN: Logical File Name
  - This could be any universal file name, not necessarily referring to the LFC or any catalog
- SURL: SRM URL
  - Specific to SRM (# from tURL as SRM is not a protocol)
  - srm://<endpoint>:<port>/<SAPath>/<path>
- tURL: transport URL
  - URL used by an application for getting access to the file
  - <protocol>://<endpoint>:[<port>]/<SAPath>/<path>
- Redirector:
  - Any service that, if necessary, redirects the request to another more suitable service
    - http proxy
    - xroot redirector

LHCb usage of an interface

- File access and transfer
  - Open files from applications
    - Uses ROOT and its specific plugins depending on protocol
  - Transfer files
    - FTS or lcg-cp (gridftp for the time being)
    - Possibility to implement plugins for native transfers: dccp, rfcp, xrdcp, cp
      - Only possible to/from local file or between same SE type
- Tape recall
  - bringOnline and cache pinning
- Selecting destination service class
  - Currently using SRM spaces (a.k.a. space tokens)
- Get information about service class status
  - Available space, used space, free space…
- Currently use SRM
  - Also rfio plugin (for pit transfers until last year)
  - dCache plugin decommissioned
    - See next slide

Why use SRM?

- Alternative: URL translation
  - In most cases, just very easy…

    `srm://srm-lhcb.cern.ch/castor/cern.ch/grid/lhcb/buffer/lhcb/LHCb/Collision12/EW.DST/00018980/0001/00018980_00012377_1.EW.dst`
  - … but sometimes inefficient
    - OK if there is a redirector for the selected protocol
      - `root://castorlhcb.cern.ch//castor/cern.ch/grid/lhcb/buffer/lhcb/LHCb/Collision12/EW.DST/00018980/0001/00018980_00012377_1.EW.dst?svcClass=lhcbdisk`
    - No redirector for gridftp…
      - `gsiftp://lxfsrb4204.cern.ch:20390/c9960884-2c3d-27a0-e043-4aa18a89bd83`
    - No redirector for dcap doors
  - Access to different services classes not easy if served by a single service
    - Put/get operations into/from a service class
      - `rfio://castorlhcb.cern.ch:9002//castor/cern.ch/grid/lhcb/buffer/lhcb/LHCb/Collision12/EW.DST/00018980/0001/00018980_00012377_1.EW.dst?svcClass=lhcbdisk&castorVersion=2`
  - Endpoints other than SRM are not public
    - gridftp servers
    - dcap doors
- SRM was supposed to work and scale!
- These are the reasons why we abandoned URL translation

Workarounds

- Tape recall & pinning
  - No alternative that I know of for many implementations
  - Now talking only about disk-only SEs
- Gridftp server
  - Use a single server with high bandwidth
    - Bottleneck for all transfers
  - Use a DNS balanced alias
    - No optimisation possible
  - In any case the server name/alias must be stable
  - Is redirection possible?
- File open
  - Redirector or similar
    - xroot, rfio, file
  - Multiple doors (dcap, gsidcap)
    - Define a DNS load balanced alias
    - Server name must be stable
- Multiple service classes (SRM spaces):
  - Deploy one endpoint per service class (site)
  - For LHCb: require 3 service classes (T1D0, T0D1 for prod and users)

String manipulation

- From the LFN, LHCb builds a SURL
  - Can easily build a tURL as well
  - Only string manipulation (but just happens it is so, not even an agreement)
- … but …
- Each experiment is implementing its method
- Each experiment has to maintain a configuration
  - Server name, port number (if needed), SA path etc…
- Why not solve only once the problem?
- B.t.w.: why was SRM (and DM middleware at large) such a failure?
  - This should have been the simplest thing to implement
  - No need for all fancy SRM

What for should we keep SRM though?

- For custodial storage:
  - Any alternative?
    - Recall, pin, unpin
- For online (disk) storage:
  - How to get real time storage usage?
  - Define a simple service interface for that?
  - Implement our own on VOBOX (if there is a local information)
    - Stager_qry -s
- Directory handling: mkdir, rmdir
  - Supported automatically by all protocols?

- Some goody of SRM spaces that none talks about
  - Some implementations (dCache) allow dynamic change of allocated space
  - No alignment with disk server pools
  - Limited but interesting usage…
  - … probably the gift compensating files outside an SRM space

Conclusion

- In many areas, LHCb could get rid of SRM
  - Requires some agreement with sites
    - Scalability
    - Stability of server names (load balancing)
    - One server per service class (avoid space tokens)
  - For T1D0, is there any alternative?
    - Of course if there is no need for tape recall
    - But this costs a lot of disk…
- Should each experiment redo the work?
  - Probably yes, as then it is integrated in their framework
    - Not LHCb viewpoint, just experience…
  - … and otherwise it would take 3 years (minimum) to (not) be deployed…
    - … remember Mumbai workshop was 6 ½ years ago and SRM v2.2 MoU (never implemented) was 1 year later…