CTA+LHCb Discussion

Minutes of the meeting of 4 July 2018

Present

- **IT:** Michael Davis, Julien Leduc
- **LHCb:** Joël Closier, Christophe Haen

Protocols

One of the key points from the discussion is that LHCb are staging files directly from CASTOR to other Storage Endpoints (SEs) across the grid. For this use case they need to support all the protocols in use at all Tier-1 sites, not just the protocols in use at Tier-0. Some sites use dCache, which does not support 3rd Party Copy (TPC). Another system in use at some sites is StoRM. (And DPM? Not mentioned in the discussion).

For EOS disk access, they use only XRootD for reading and writing but for CASTOR tape they use only SRM, for the reason mentioned above: their software needs to write to SEs which don't support XRootD, and they want to use the same protocol everywhere.

The main library/abstraction layer that they use to query stagers *etc.* is `gfal2`. They use the Python bindings to the library.

They have also started to add support for ECHO (Ceph-based disk storage at RAL). RAL prefer to use `GSIFTP` with an XRootD plugin written by Sebastien Ponce. `GSIFTP` is a subset of the `GridFTP` protocol, essentially standard FTP enhanced to use GSI security. It does not include many of the high-performance `GridFTP` protocol features, such as parallel data transfer, automatic TCP window/buffer sizing, enhanced reliability, *etc.* LHCb will need to be able to write to RAL using TPC.

⚠ Find out from Giuseppe how it currently works when staging from CASTOR to ECHO disk at RAL.

From the pit, they write directly to CASTOR and data is immediately staged for writing to tape.

TPC must be possible to all SEs not just T0. This will be made possible using XRootD v5 which is due to be released in the autumn.

If we don't want to allow files to be staged directly from CTA to Tier-1 SEs, this will require a change in LHCb workflows, so we will need to discuss this with them.

Space Token

The Space Token is a SRM concept. Many sites only work with the Space Token so are bound to SRM.

The Space Token is also used for accounting. However, on CASTOR space accounting gives the space used on the disk stager, not the amount of data archived to tape, so this is not so useful.

⚠ Is this the same thing as "JSON file with space" that ATLAS were talking about?

Permissions

The typical use case is production staging. User staging is very rare. However, in some cases users need permissions to access tape (batch retreives).

In CASTOR, only the Production role in LHCb VOMS has the rights to write to tape, but some users without that role need to have permission to read. One common use case is that their calibration guys need to recall a single file. They are a special group who have this specific requirement. They don't need write access.

LHCb say that they don't do the thing with multiple requests for the same data and then cancelling them as soon as one is fulfilled. If they request to stage something, that means they really want it.

Permissions need to be consistent across all WLCG SEs. Most users are not CERN users.

There was a question about how grid certificates will be mapped to privileged users. Will this be under the control of the experiment or will they have to ask us to do it?

Testing

LHCb are happy to let ATLAS do the load testing. If CTA can handle ATLAS loads, then it will not have a problem with LHCb loads.

A good test for them would be to stage 100,000 files from tape and distribute them to many sites across the grid.

Additional Notes from Christophe 17/07/2018

There are two kinds of jobs: production jobs and user jobs.

Production Jobs

These are run by the experiment's data management experts, who control what jobs run and when they run.

The normal use case is pre-staging the data: copy all the needed data to a disk storage (T0 EOS or another T1 SE). Apart from some exceptional circumstances, production jobs do not access tape storage. The output is always written to disk. The transfer to a tape storage is done asynchronously using FTS.

User Jobs

Users are allowed to run jobs on files in CASTOR (with some restrictions). Depending on the computing model that will be adopted for Run3, this might increase. At the moment it is “very low”.

When a user job requests a file on tape, before starting the job, the Python/gfal2 programs are used to centrally stage the required files. Once the file is in the CASTOR disk cache, the job is woken up and sent to a site, which will try to read the file from the disk frontend.

Output of user jobs are never written to tape.