

## Experience with SRM 2.2 (Sophie Lemaitre)

There is one endpoint for SRM 2.2 for each backend. Clients are being written. This include Flavia's functionality tests, lcg\_utils, GFAL, FTS. Looking at Flavia's functionality tests, a lot of progress have been made in the last few months.

In the test for "ping" and the one of "put cycle" all backends perform near 80% or more, except for CASTOR, where the success is about 40%. The CASTOR problem seems to be understood, developers are working on it.

Summary of the backend status from presentation and discussion: the DPM and dCache implementations have shown good stability and availability in the last period. CASTOR still shows some issues, but a lot of progress has been made so far. However, the tests have been limited to basic functionality tests, there have been no test of experiment specific use case. Moreover, there has been no stress test so far. The impression is that still a lot of work will be needed to provide stable endpoints.

Status of the clients: FTS clients are in the testing phase before release, lcg\_utils and GFAL are under development and testing. The main issue is the different behavior of the various implementations for the same method: consistency between implementations improved a lot, but still the behavior is not always unique.

Deployment scenario: SRMV1.1 and SRM2.2 will need to coexist until SRM2.2 is not proved to be a stable service. In the coexistence period, the default for the clients will be SRM 1.1. In the discussion it was made clear that this coexistence implies only duplication of SRM frontend, no storage backend or data replication.

## Castor and SRM 2.2. (Luca dell' Agnello):

Currently there are 4T1s with CASTOR installations but also T2 might use it w/o tape backend (disk only). There are different versions of CASTOR ad different T1.

T1s sholud upgrade to SRM2.2 in the first quarter (ready for production in may). CERN and RAL already providing a test endpoint, CNAF in starting the installation process.

At CNAF, D0T1 will be castor. D1T1 probably castor. D1T0 could be CASTOR, but other possibilities are being investigated (Storm/dCache). This is possible, since T1D0<->TOD1 transitions are not foreseen.

There has been a discussion about TOD1 in Castor. From the discussion it emerged that both CNAF and CERN provide TOD1 in fact as T1D1, which means they basically turned off the GarbageCollector. This is done for operational reasons i.e. it is the simplest and more reliable way to provide a CASTOR SE with a permanent copy of data on disk. It also offers a more reliable storage (tape backup of the file) for a manageable cost (extra tape usage).

## SRM 2.2 in dCache (Doris Ressimann)

dCache 1.7 includes already some SRM 2.2 features. The SRM is on a separate machine, need more resources. The implementation of 2.2 features is almost ready, there are 4 open issues, 2 of easy solution, the other 2 require more work.

dCache 1.8 will be ready for deployment in April. The update plans are still not clear. There some info needed from the experiments and need to have the 1.8 version tested to understand the plan.

BNL will upgrade to 1.7 next week. Would like the “overwrite feature” not in 1.7. PIC is using dCache for disk, CASTOR for tape. TRUIUMF will upgrade to 1.7 in 2 weeks. NDGS have a distributed setup across different sites (challenging). FZK is already running 1.7 since January. IN2P3 upgraded in November and solved some initial problems.

There was a question about the statement that for D0T1 in dCache the idea is to separate input and output buffers. It was not clear what this implies: if the experiment writes data in the input buffer, will they need to pass them to tape before reading them in the (read-only) output buffer? The answer was NO: dCache allows to flush them from one pool to the other. For the user is transparent, it is an implementation detail.

## SRM 2.2 in DPM (Marco Serra)

SRM2.2 for DPM is not deployed in INFN yet, so the discussion will be on the current version of SRM and the current situation. At the moment 81 DPMs are published in the Information System. 61% of them publish wrong information (both zero used and free space, or PB of available space). There are only few installation with a reasonable amount of free space.

Points from site administrators: the support of SLC4 and 64 bit architecture is crucial. It would be useful to have tools for consistency checks in DPM, verifying the files physically on disk against the entries in the namespace. There is the need of monitoring tool (both real time and static) for the farms, with different views (per VO, per pool, ...). Other wishes include implementation of VO-based quotas, a rate-limiter to avoid DOS attacks, support for Xrootd (ALICE sites).

CMS reported problems with the performance of DPM, due to the size of the DPM DB. A system for cleaning old put/get requests is needed. At the moment the problem has been patched creating DB indexes.

ATLAS had permission problems using DPM with different VOMS. For the short term a fix has been provided, but support for secondary groups is seen as an urgent necessity.

The problem of incompatibility DPM/Castor RFIO library is still there. It seems more a coordination problems rather than DPM problem, but the fix has not been integrated in the release yet.

From the presentation, discussion and the survey emerged that DPM is easy to install and maintain, but some extra tools are needed. The main concern is the scale of DPM: it is not clear if DPM has ever been tested at the right scale, probably not. Therefore we do not know its limits and when they will be hit. Also, it is not clear if the new features will arrive in time for the challenges T2s have to face. The limited amount of manpower dedicated to DPM development (2 FTEs) seems a problem.

SRM2.2 in Storm (Luca Magnoni)

Sotrm is an INFN SRM implementation. It supports GPFS, XFS and any file system with ACL support. It enforces ACLs on physical files. Storm can access the storage on behalf of the user or the application. Migration to 2.2 is complete and all functions requested by WLCG are implemented.

#### DISCUSSION:

There has been a lot of discussion initiated by a statement from RAL: "For the moment at RAL it is not foreseen to support storage classes transitions. Experiments should be aware of this". Maarten Litmaath commented that T1D0 -> T1D1 transition needs to be supported for reprocessing data (data need to be brought online for the period of time of reprocessing). From the RAL side, there is some intensive testing needed to understand if and which such transitions can be allowed. This is not only an implementation constrain but also involves resource allocation of the site. The conclusion is that it is not crucial if the T1D0 -> T1D1 transition will not be available for a while, but it should be made possible at some point before the end of the year.

From the T2 side there is the problem of space reservation: in many cases, one single VO reserves/uses all the space in the SE. There is the need of space allocation per VO in the SEs.

Concerning the clients, it emerged that GFAL/lcg\_utils will support space tokens.

It seems that some sites encountered performance issues with DPM and decided to move to dCache after SC4. DPM developers pointed out that such problems should be reported

to DPM developers. Performance issues need to be tracked down and investigated. The developers received only few reports of performance issues.

The last part of the discussion focused on what experiments should expect in terms of reliability from the storage and requested service. Alice made the point that the experiment has provided the amount of space needed, for the reliability, it should be closed as much as possible to 100% and indicating matrices for reliability is not reasonable. In respect of functionality, the sooner new feature are in place, the better it is. After a bit of discussion, it was agreed that the experiment could indicate what to expect in terms of stability of service and scalability for their exercises in the near future (next 4 months?). There is a group for Storage Classes where all this is discussed.

Last comment was that developers should not forget there are other VOs which are not HEP.