

# Table of Contents

<b>LHCb input to "SRM and Clouds" in the Storage TEG.....</b>	<b>1</b>
<b>SRM usage in LHCb.....</b>	<b>2</b>
Ability to identify different storage classes (SRM spaces).....	2
Space usage accounting.....	2
Alternatives.....	2
Get a tURL from an SURL.....	2
Alternatives.....	3
Handling custodial storage.....	3
Alternatives.....	3
Conclusion.....	3

# LHCb input to "SRM and Clouds" in the Storage TEG

This page summarizes the LHCb understanding of the "SRM and Clouds" topic discussed within the WLCG Storage and Data Management TEGs.

The title "SRM and Clouds" is misleading. The alternative probably (but I am not sure of what Ian meant) is between "classical HEP storage" (Castor, dCache, DPM, StoRM) and "storage with cloud interface". Although cloud storage may be coming at some small sites, and should not be neglected, the main issue is how to use it (mostly disk-only probably) and how to transfer data from an HEP storage to it.

I'll now concentrate on how to interface "classical HEP storage".

# SRM usage in LHCb

Currently SRM is widely used (but not only). However one should not make confusion: SRM is *not* an protocol for storage access, but a service that allows to interface to storage. Each storage has its access protocols (for POSIX-like and transfer access, mostly `gsiftp`). The main reasons why SRM is used and required are the following:

1. This is the only protocol that FTS understands.
2. it provides the minimal functionality (but also much more) that is required for our data management activities

Let's review here the functionality of SRM that is used and how it could be replaced or worked around if we don't use SRM any longer.

## Ability to identify different storage classes (SRM spaces)

Currently none of the SRM space handling functionality is used. Spaces (a.k.a. space tokens) are created by site managers on request and in agreement with VO data managers. The spaces allow to identify the required storage class when uploading a file to the SE and (for Castor only) when recalling a file from tape. Depending on the implementation there may or may not be an SRM-level accounting of space used (yes for dCache, no for Castor, no for StoRM).

Besides this capability, the only service call concerning spaces is to get the available and used space (space accounting)

## Space usage accounting

This is used by Data Managers to assess the occupancy of the disk space. It returns the available and used space. In LHCb we use it to generate SLS plots, like this one.

## Alternatives

An alternative to using spaces is to use multiple endpoints, or multiple ports on the same endpoint. Another one is to link the service class to the namespace of the files. Although this was highly discouraged initially (since SRM was supposed to provide the ability to move files between spaces), this is no longer relevant and one could envisage an SAPath for each service class, e.g.

```
/grid/lhcb/tape  
/grid/lhcb/disk
```

Migration of existing files to such a system may be difficult, and should therefore be carefully prepared and handled mostly by the site managers, using more powerful tools for renaming files.

## Get a tURL from an SURL

Most used functionality of SRM probably. It allows to get a handle to a file for a given protocol, or for the most suitable protocol provided in an ordered list. Users therefore don't need to know which are the protocols supported by the local installation. Providing a list of protocols like:

```
file,xroot,root,dcap,gsidcap,rfio
```

allows to get a tURL for opening the file in the most suitable way using ROOT. Of course the order is a matter of taste.

The same functionality is used internally by FTS to get a gsiftp tURL.

## Alternatives

Our framework can create a tURL, knowing the following information:

- Endpoint and port number (if needed)
- SAPath (i.e. base directory)
- Path (in our case the LFN)
- Service class (if needed, e.g. for Castor)

There is however a caveat, which is that the endpoint is not necessarily publicly known, and for practical reasons there should be a single endpoint per SE. This may result in this endpoint (used therefore as a redirector) to be overloaded depending on the implementation (e.g. dcap doors). For technologies that anyhow use a redirector (Castor, xroot), this is not a problem. A solution would be to publish an endpoint as a load-balanced DNS alias. What is important is that it is unique and stable.

## Handling custodial storage

SRM is the only way to recall files from tape for some implementations, in particular dCache. In case the file is accessed directly on the disk cache, it is mandatory to have (soft or hard) pinning and unpinning capabilities.

## Alternatives

Manual requests for massive staging is not considered as a viable alternative!

In case a service is provided for moving files from a T1D0 service class (including suitable recall from tape) to a T0D1 service class, one can consider to replace pinning with a replication (i.e. including registration in a replica catalog). Unpinning would then consist of deleting the replica. For example if FTS includes this capability, whatever the mean is for handling pre-staging, a replication may replace bringOnline and pinning capability of SRM. Note however that this requires additional T0D1 buffer space which could partly come from a decrease of the T1D0 cache buffer. This functionality should be a bulk transfer request, and user should be able to know which files are available at any time, in order to schedule jobs.

We recognise this consists of pushing part of the SRM functionality onto FTS and the experiment's framework (for removing replicas), but this is certainly possible.

## Conclusion

With minimal developments and agreement on operational matters, it is certainly possible to avoid using SRM.

The main development concerns "FTS", i.e. a reliable and monitorable file replication facility providing as a service "transfer this dataset to this SE, and let me know which succeeded and which failed". Non fatal failures such that system overload should not even be returned to the user but handled internally.

Operational agreements are:

- Define one endpoint per storage class
- Define DNS load-balanced aliases for endpoints for protocol servers (if needed)

-- PhilippeCharpentier - 02-Mar-2012

This topic: LHCb > SRMUsageInLHCb

Topic revision: r1 - 2012-03-02 - PhilippeCharpentier



Copyright &© 2008-2019 by the contributing authors. All material on this collaboration platform is the property of the contributing authors.

Ideas, requests, problems regarding TWiki? Send feedback