Storage Resource Reporting

Proposal for storage providers

v3.0 11/10/2017

Alessandro Di Girolamo, Oliver Keeble

Table of Contents

1 Introduction

This document summarises a proposal for storage resource reporting in WLCG, intended to enable experiment operations and WLCG storage accounting. The document proposes five requirements, four of which are targeted at the WLCG storage providers (dCache, DPM, EOS, StoRM, XRootD). This document has been produced in consultation with all the LHC Experiments..

2 Proposal

We start in section 2.1 by documenting what minimal resource reporting functions *any* storage service used by WLCG experiments would need to provide. This is intended to cover all services of interest, including those provided by commercial cloud providers, or those based on third party technology such as Ceph. This is documented as requirement **R0**.

This rest of the document is oriented towards the "WLCG storage providers", in other words storage systems which will respond to the community's requirements in order to allow better integration into experiment frameworks. As such, these requirements can be interpreted as "requests" in the sense that systems are in principle still usable by the experiments without them.

Sections 2.2 and 2.3 are the core, and represent the ability to query resource usage information from storage systems without the use of SRM. Taken together, they would enable the proposed WLCG storage accounting system. Subsequent sections relate to a withdrawn requirement on subdirectory reporting and to the provision of storage dumps.

2.1 General Requirements

REQUIREMENT R0

Storage services must provide

- Total used space.
- List of files stored (no other metadata required)

The total used space is needed for high level accounting, it can also be used to estimate any discrepancy with the VOs accounting and therefore if there is significant amounts of dark or lost data. The list of file stored, will allow for a consistency check to be run.

Note that some cloud providers do not have the concept of free space – more can be purchased at any time. If the json file concept is accepted, a number could be inserted 'manually' into this file to set a storage limit. This would only be a soft limit but would allow the calculation of a 'free space'.

The file list is required for those systems whose interface operates at this level (e.g. file access, or object access where an object represents one or more files). Lower level services such as block storage devices are not covered here. They may still, however, contribute to storage accounting by advertising their occupancy through the mechanism described in **R2**.

2.2 "Space quota" level resource reporting

The entities queried are the same spaces which are currently referred to as space tokens, or simply "spaces", here referred to as "*space quotas*".

REQUIREMENT 1 (R1): Storage systems should provide total used and total free space for all distinct *space quotas* available to the experiment through a non-SRM protocol.

This information is intended to serve (at least) two use cases

- determining the total capacity used and available to the experiment
- allowing the experiment to manage its occupancy of the storage to avoid filling it up and to maintain any targets of available space (e.g. by selective deletion).

Other requirements.

- Query frequency – order of minutes (not Hz)
- Accuracy
  - volume - order of tens of GB (i.e. experiments are not picky on super precisions, Storage providers should comment on what's doable with a limited amount of complication).

- time – a freshness of tens of minutes or so is acceptable

The accessibility of these numbers depends upon the storage system type, the protocol, and configuration decisions relating spaces quotas with the namespace. In reality, there are three possibilities, gridFTP, HTTP and xrootd. A storage system should implement resource reporting in at least one protocol. NB – Alice request that all supported protocols provide this capability.

The interfaces are described in the subsequent sections.

2.2.1 GridFTP

SITE <sp> USAGE <sp> [TOKEN <sp> $token <sp>] $path

which returns

250 USAGE <sp> $val FREE $val2 TOTAL $val3

- Space usage in bytes (required)
- Available space in bytes (required)
- Total space in bytes. This is optional: if not provided, the total is assumed to be the sum of the used and available space.

https://github.com/bbockelm/globus-gridftp-osg-extensions

2.2.2 HTTP

This requires association of space quotas to the namespace and implementation of RFC4331.

$ echo -e '<?xml version="1.0" ?>\n<propfind xmlns="DAV:"><prop><quota-used-bytes/><quota-available-bytes/></prop></propfind>' | curl -sL --capath /etc/grid-security/certificates --cert /tmp/x509up_u1000 -X PROPFIND -H "Content-Type: application/xml" -H "Depth: 0" -d @- https://<host>:<port>/<path> | xmllint --format -

which returns

[...]

<lp1:quota-used-bytes>24722066801</lp1:quota-used-bytes>

<lp1:quota-available-bytes>75277933199</lp1:quota-available-bytes>

[...]

2.2.3 xrootd

The following are (equivalent) examples of querying space through the xrootd command line

```
$ xrdfs <endpoint> spaceinfo <path>
$ xrdfs <endpoint> query space <space>
```

2.2.4 Provider Summary

Storage	Version	GridFTP	HTTP/DAV	Xrootd	Namespace association *
dCache	> 3.2	YES	YES		Possible
DPM	>1.9.0	NO	YES	COMING	Obligatory
EOS			YES**	YES	
CASTOR				YES	Not relevant
StoRM			YES	Not directly supported by StoRM itself, but possible	?
xrootd				YES	Possible

Table 1

Table 1 summarises support for space reporting in the various storage systems.

* The namespace association column is intended to indicate whether the spaces managed by the system are orthogonal to the namespace (SRM style) or can/must be associated with a path.

** An “empty” PROPFIND will not list the properties but they can be queried directly.

2.2.5 Clients

The relevant numbers will be made available through the gfal2 interface to extended attributes (gfal2_getxatt etc.)

2.3 Summary topology files

REQUIREMENT 2 (R2): provide a summary file indicating the “topology” of the system

Atlas have proposed that a summary of the storage system (aka “the json file”) be made available, either within within the namespace of the system itself or on a separate system. An example file is given in Appendix A.

In order to use this information to deduce the total resources allocated to the experiment, it must be valid to sum the totals, i.e. the spaces must be independent, and the list must be complete.

The json file has the following advantages

- it solves the “topology problem” of knowing which spaces to query in order to assemble a complete view of capacity
- it can be cached to allow high query rates
- by setting permissions appropriately, it would avoid an access problem; with which what credential should a central service query experiment space usage via standard protocols?

The freshness requirements received indicate that periodic generation of this file would be acceptable.

For the WLCG accounting use case, the file’s path and name are up the site as the system will be bootstrapped via GOCDB or OIM. This system even envisages supporting two files, one with static topology and one with dynamic occupancy data. Experiment policy may dictate that a copy should also reside at a well known place in the experiment area.

This proposal assumes that it is acceptable that this file be world readable, as the accounting system will in any case make the data public.

2.3.1 Tape

Relevant numbers should be available also for tape systems, which typically comprise a tape backend and a disk cache. The disk cache can follow reporting guidelines above, whereas the tape part could be published by the topology file. This is likely to be for used space only (NB Alice are interested in “free space” too, to understand if they can successfully write). Aggregations would not be based on path but some other meaningful grouping, e.g. tape families, or could simply be a single number per V.O.

2.3.2 Generic Storage

As the proposed summary file is based on the Glue2 schema, it is intended to be flexible enough to allow publication of resource use for generic services which are not storage systems in the traditional sense. For example, a large Elasticsearch service could publish storage accounting information this way.

2.4 Subdirectory level resource reporting

This requirement has been **withdrawn** as no experiment supported its inclusion. For reference, the original text read;

> **REQUIREMENT 3 (R3)**: Provide used and free space on subdirectories, in particular any entity on which a *restrictive quota* has been applied. This will allow clients to understand if they can write.

Note that the withdrawal of R3 implies that the granularity of the resource reporting is tied directly to the compartmentalisation of the storage system into separate spaces.

2.5 Storage dumps

REQUIREMENT 4 (R4): Provide full storage dumps

Full storage dump enumerating each file. The aim is to allow a single utility per storage system which will work for all interested experiments. The following information represents the union of the attributes requested by Atlas, CMS & LHCb.

- path
- size

- creation time
- checksum type & value
- start and end timestamps for the dump
- last access time - optional

To be provided on weekly timescale when not possible with higher frequency (e.g. EOS find allows, in a couple of hours, of having the full dump. CMS run it daily).
A cross-experiment tool supported by the storage provider is requested.

While it is beyond the scope of this document, it should be noted that in order for experiments to make use of these storage dumps (e.g for consistency checking), they will need to provide the same information for the files they believe are at the site. If the experiments could make this available to the sites in a consistent manner, then this would allow for the development of tools that are independent of any experiment software.

3 Usage for WLCG Accounting

The proposed WLCG Storage Accounting system was described at the accounting WLCG Accounting Task Force Meeting on 20th April 2027 (https://indico.cern.ch/event/632531/). It relies on requirements R1 and R2 from this document.

Only the summary information described above is required for accounting. The workflows of the experiments related to space accounting won't change, but agreeing on the format, structure and content of the information exposed will enable the possibility to setup a WLCG collector in parallel to the experiments workflows to collect the Storage Resources accounting information.

More information on WLCG storage accounting can be found at https://twiki.cern.ch/twiki/bin/view/LCG/AccountingTaskForce#Description_of_the_storage_topol.

4 Overview of input received

The following table summarises which information is required, interesting or not required for each stakeholder.

	Summary info	Subdir reporting	Detail storage dump
Alice	Via all supported protocols		Not required, provided find/ls works on Xrootd protocol
Atlas	Via at least one protocol or via json		Path. Optional - size, atime. Once per month
CMS	Via at least one protocol, or json if protocol is not possible		Path, size, checksum
LHCb	Via at least one protocol protocol, json (?? tbc)		Path, size, ctime. Once per week
WLCG	Requires json file		Not required

Table 2

5

6

7 Appendices

7.1 Appendix A

As part of a larger initiative focused on the evolution of the WLCG information system, a Glue2 inspired description of a storage system, rendered in json, is being formulated. In order to remain compatible with future developments in this area, we here propose the strict subset of this description which is required to support storage resource reporting. The example here should be read in conjunction with the full (and evolving) description of the schema, linked from https://twiki.cern.ch/twiki/bin/view/LCG/AccountingTaskForce#Description_of_the_storage_topol.

```
{
    "storageservice": {
        "name": "storage.org",
        "storageendpoints": [
            {
                "interfacetype": "gsiftp",
                "assignedshares": [
                    "all"
                ],
                "endpointurl": "gsiftp://storage.org/"
            },
            {
                "interfacetype": "https",
                "assignedshares": [
                    "all"
                ],
                "endpointurl": "https://storage.org/"
            }
        ],
        "storageshares": [
            {
                "name": "TOKEN01",
                "totalsize": 100000000000,
                "timestamp": 1504699646,
                "assignedendpoints": [
                    "all"
                ],
                "usedsize": 3385731314,
                "numberoffiles": 1231325,
                "groups": [
                    "atlas"
                ],
                "path": [
                    "/top/vo/atlas"
                ]
            }
        ]
    }
}
```

Notes

- `numberoffiles` is requested but optional
- `path` may not be relevant and is optional
- the full schema includes many other entries which storage providers may wish to include. Here, we only list those required for resource reporting