Enabling Grids for E-sciencE

Storage Classes summary of deployment discussions

Flavia Donno

Pre-GDB - CERN, 9 January 2007

www.eu-egee.org

Information Society and Media

Outline

- Storage Classes
- Storage Class Transitions
- WLCG Storage Element Model
- Miscellaneous items
- Summary of Discussions
- Plans
- Summary of Implementations

Storage Classes

- A **Storage Class** *determines the properties that a storage system needs to provide in order to store data.*

- The LHC experiments have asked for the availability of combinations of the following storage devices: **Tapes** (or reliable storage system always referred to as tape) and **Disks**.

- TapeN and DiskN:
  - **If a file resides on Tape then we say that the file is in Tape1.**
  - **If a file resides on an experiment-managed disk, we say that the file is in Disk1.**
  - **Tape0 means that the file does not have a copy stored on a reliable storage system.**
  - **Disk0 means that the disk where the copy of the file resides is managed by the system: if such a copy is not pinned or it is not being used, the system can delete it.**

Storage Classes

- **Supported Storage Classes** *for next implementation of SRM v2.2 are:*
  - Custodial-Nearline: this is the so-called Tape1Disk0 class.
  - Custodial-Online: this is the so-called Tape1Disk1 class
  - Replica-Online: this is the so-called Tape0Disk1 class

  - Tape0Disk0 is not implemented. It is pure scratch space that could be emulated using one of the available classes and removing the data explicitly once done. However, it could be handy for LHC VOs to have such a type of space actually implemented.

Storage Classes

- ***Custodial-Nearline***: data is stored on tape. Access to data may imply certain latency. When a user accesses a file, the file is recalled in a cache that is managed by the system (Disk0). The file can be "*pinned*" for the time the application needs the file. However, the treatment of a pinned file on a system-managed disk is implementation dependent, some implementations choosing to honor pins and preventing additional requests, others removing unused on-line copies of files to make space for new requests.
- ***Custodial-Online***:data is always available on disk. A copy of the data resides permanently on tape. The space owner (the virtual organization) manages the space available on disk. If no space is available in the disk area for a new file, the file creation operation fails.This storage class guarantees that a file is never removed by the system.
- ***Replica-Online:*** it is implemented through the use of disk-based solutions not necessarily of high quality. The data resides on disk space managed by the virtual organization.

Storage Classes Transitions

- Through the SRM call srmChangeSpaceForFiles it is possible to schedule Storage Class Transitions for a list of files.
  - **Tape1Disk1 -> Tape1Disk0**.
  - **Tape1Disk0 -> Tape1Disk1**. This transition would be implemented with some restrictions: the srmChangeSpaceForFiles call will complete successfully but the files will remain on tape. The files will be actually recalled from tape to disk only after a BringOnline operation is executed. This is done in order to avoid that a big set of files is unnecessarily scheduled for staging and therefore to smoothen operations in particular for those Mass Storage Systems that do not have a scheduler (namely TSM).
  - **Tape0<->Tape1** transitions are **not supported** at the start of LHC (if ever). For physics validation operations, since the amount of data to transfer to tape after the validation is not big (only 1-2% of total data) a change class operation from Tape0Disk1 to Tape1DiskN can be approximated by copying the files to another part of the space, specifying Tape1DiskN as the new storage class, and then removing the original entries.

WLCG view of SE Model

- A **Storage Element (SE)** *is an aggregate of Grid services that allows Grid users to store and manage files together with the space assigned to them*.

- The **SE Implementation** is the software system used to manage the storage devices and servers. Examples of this are: CASTOR, dCache, DPM, StoRM, etc.

- An SE exposes **Total Sizes**: an **Online** for space on disks and **Nearline** for space on tape or slow devices.

- An SE can have multiple **Storage Areas**.

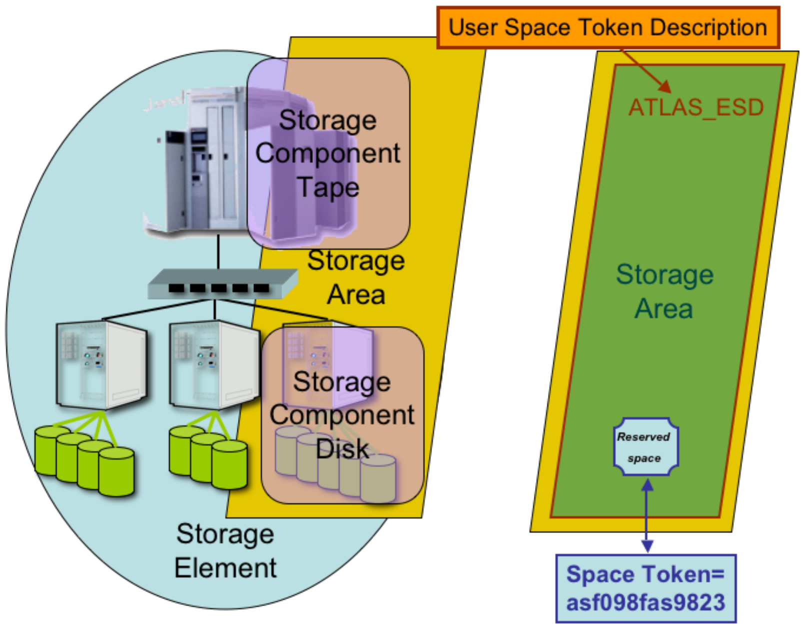
The Storage Area

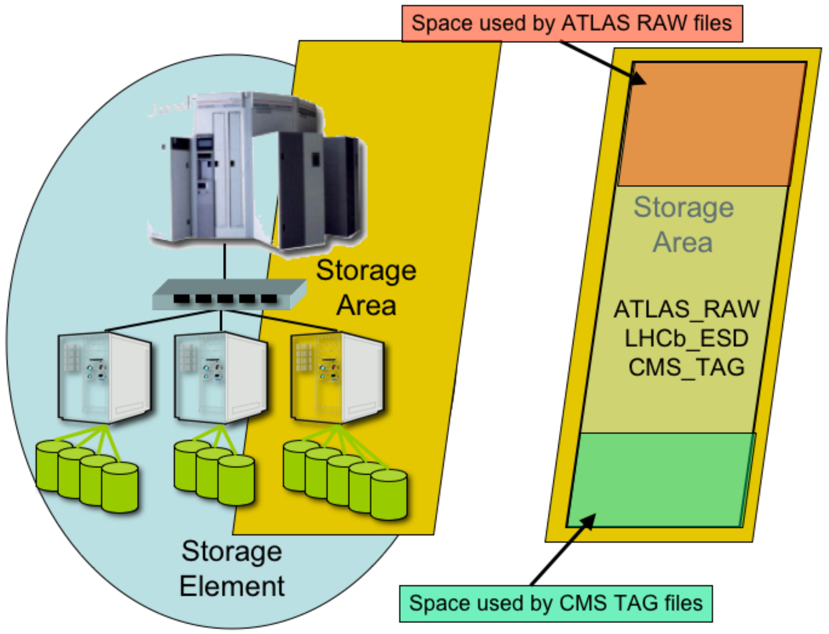
- A **Storage Area (SA)** *is a view on a portion of the total space:*
  - *It is created by the System or VO Administrators*
  - *It can span different kinds of storage devices within a Storage Element*
  - *It exposes a single retention policy and a single access latency (which the underlying storage devices together can support)*
  - *In case of WLCG it implements a Storage Class instance.*

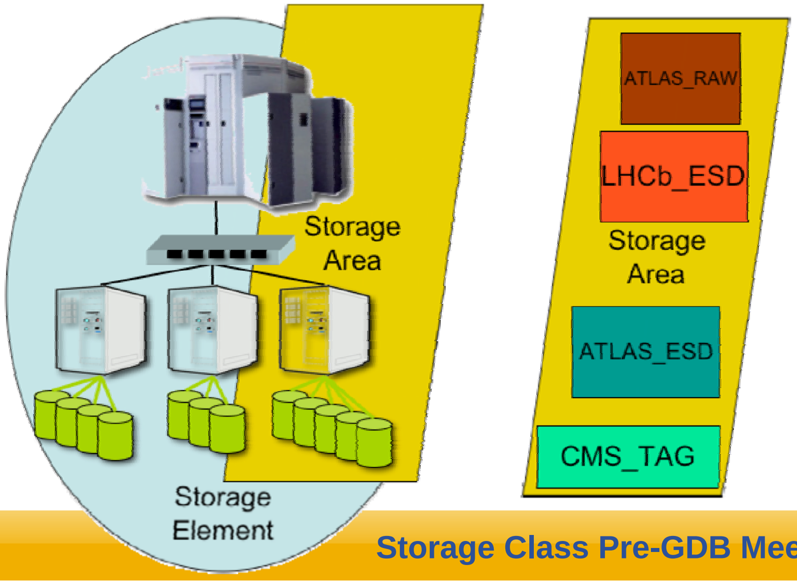
- An **SA** may be dedicated or shared between certain Vos/groups/roles
  - **For WLCG the default SA is typically shared** (it is Tape1Disk0 for CASTOR at CERN)

- For WLCG the SA implements a Storage Class instance:
  - **It is identified by a** Space Description **(that can differ per VO).**

The Storage Area







Miscellaneous Items

- Clients should *not use dynamic reservation* initially.
- *Storage Areas are created statically* by Site Administrators at sites. VO administrators can use srmReserveSpace to create Storage Areas.
- *Tools will be available to publish VO specific Space Descriptions (ATLAS_RAW, CMS_TAG, etc.)*
- *Namespaces are meaningful for certain implementations such as dCache and StoRM.*
  - dCache uses paths to specify dCache Storage Groups (SGs identifies tape sets or other space characteristics for VOs).
  - In dCache, experiments will be able to change Storage Classes without having to change the path for a given set of files.

Summary of discussions

- The concept of the Storage Area was somehow considered difficult to adapt to existent concepts in the various implementations. However, after some discussions good compromises have been found.
  - **For dCache it maps to pool groups/link groups: it is advised not to have overlapping Storage Areas/shared by several VOs.**
  - **For DPM it maps to filesystem pools (internal tag needed)**
  - **For CASTOR SAs are dedicated to VOs and identified by the VO specific Space Tokens. They map to Service Class Instances**
- The Glue v1.3 schema for the SE ready for deployment by the end of January 2007. Static Information providers in YAIM ready by end of February 2007. Developers are encouraged to produce Dynamic Information Provider by the same date. Implications on deployment.
- The Namespace is orthogonal to the quality of space.
  - **It could be handy to organize the namespace at all sites per VO in order to better perform operations on a set of files.**
  - **The namespace can be changed (files can be moved as long as they stay in the same Storage Class).**
  - **Experiments should agree on the structure of their namespace.**
  - **Space descriptions should be the same at all sites**
  - **The namespace is meaningful for some implementations such as dCache (the token takes precedence over the namespace).**

Summary of discussions

- Clarification for Storage Area to Tokens to NameSpace mapping
  - **Clear indications/suggestions should come from SRM developers, experiments, Glue experts**
- The possibility to move data from a space token to a new one should be provided
  - **It implies a ChangeSpace for files that are in the old space (asynchronous operation)**
  - **New files will use directly the new token.**
- Experiments should clarify how big the disk buffers should be.
  - **The amount of paid disk is specified in the MoU**
  - **Clear statements about how big transfer buffers/Disk0/Disk1 buffers should be.**
- Implementations of transfer buffers and disk/tape pools look very different at the moment at Tier-1s (IN2P3, SARA, FZK)

Summary of discussions

- Useful to have other meetings on the matter
  - To understand other T1 implementations (what is the best strategy ?)
  - To have more input from the experiments (how many space tokens ? What kind of buffers ? How many ? Namespace mapping ? ...)
  - To have clear indications from all developers (DPM, CASTOR, ...)
  - To understand implications for T2s.

Plans

- Collect requirements per VO in terms of:
  - Storage Classes needed at various sites
  - Data Flow between Tier-0, Tier-1s, Tier-2s
  - Space reservation requirements: static and/or dynamic
  - Space Token Descriptions per VO: which ? How many ?
  - Special requirements: xrootd, etc.
  - Data Access Patterns
  - Plans from 1st of April 2007 till end of the year
- Discuss with the developers on how to best implement the requirements considering the current setup of the sites (T1s and T2s)
- Produce manuals and guidelines for deployment
- Selecting sites as guinea pigs
- Testing with experiments the setup at sites
- Assisting the experiments during their tests and with possible further requirements

- Review the status of the plan at the WLCG workshop in January
- What about the new Storage Class Working Group ?

Summary of Implementations

- **GridKa:**
  - dCache+TSM + DPM
  - Separate Pool Groups per VO (T1D1, T1D0, T0D1)
    - T1D1 used for WAN T1/T2 inter-traffic and for local use.
    - T1D0 used for T0/T1 operations (48 hours, no pinning)
    - T0D1 for Grid jobs
    - Exploring PP pool set (copy through mode) available in dCache 1.7.
- **Lyon:**
  - dCache+HPSS
  - Proposal on how to map Storage Classes to dCache/HPSS Class Of Service
    - T1D0: separate buffer for incoming data to buffer for outgoing data
    - T1D0: separate buffer for RAW data to buffer for ESD
    - T0D1: collaboration with experiment to configure optimally
    - T0D1: separate production pools from user pools
    - T1D1: use dCache “file hopping” to move files from transfer to disk pools.
  - Needed experiment input

Summary of Implementations

- **SARA:**
  - dCache+CXFS/DMF
  - Tape pools are read/write/cache, disk pools are read/write. No distinction between transfer and processing pools.
  - Pools dedicated to VOs.
  - All pools are WAN pools because of NIKHEF.
  - T0D1 with non HSM dCache pools. T1D0 with HSM pools.
  - T1D0->T1D1 does not imply immediate staging of a file. However, staging requests should be known in advance in order to avoid DoS tape attacks.
  - dCache namespace used for mapping with cxfs/MSS.

References

- WLCG SRM v2.2 MoU: https://srm.fnal.gov/twiki/bin/view/WorkshopsAndConferences/GridStorageInterfacesWSAgenda

- WLCG Storage Element model for SRM v2.2 and Glue Schema Description: http://glueschema.forge.cnaf.infn.it/Spec/V13

- Minutes of the last pre-GDB Storage Class meeting: http://indico.cern.ch/materialDisplay.py?materialId=0&amp;confId=a058490