Storage Classes at Lyon Tier-1

Lionel Schwarz <schwarz@cc.in2p3.fr>

CC-IN2P3

Centre de Calcul de l'Institut National de Physique Nucléaire et de Physique des Particules

Contents

- Current implementation of storage classes
  - In dCache
  - In HPSS
  - Example of experiment use cases
- What could be done to implement SRM 2.2 storage classes

Storage classes in storage systems

- Storage Group in dCache
  - Defined by **tags** (hidden files) in the PNFS namespace
  - Linked with **pool group(s)**
  - That way, dCache knows where to physically store new files created in the namespace
- Storage Class in HPSS
  - COS (Class Of Service) is not defined in HPSS namespace
    - Associated with resources (type of disk, type of tape)
    - Associated with policies (filesize, migration, purge...)
  - Families are defined in the HPSS namespace
    - Allow to group files from a family on the same tapes

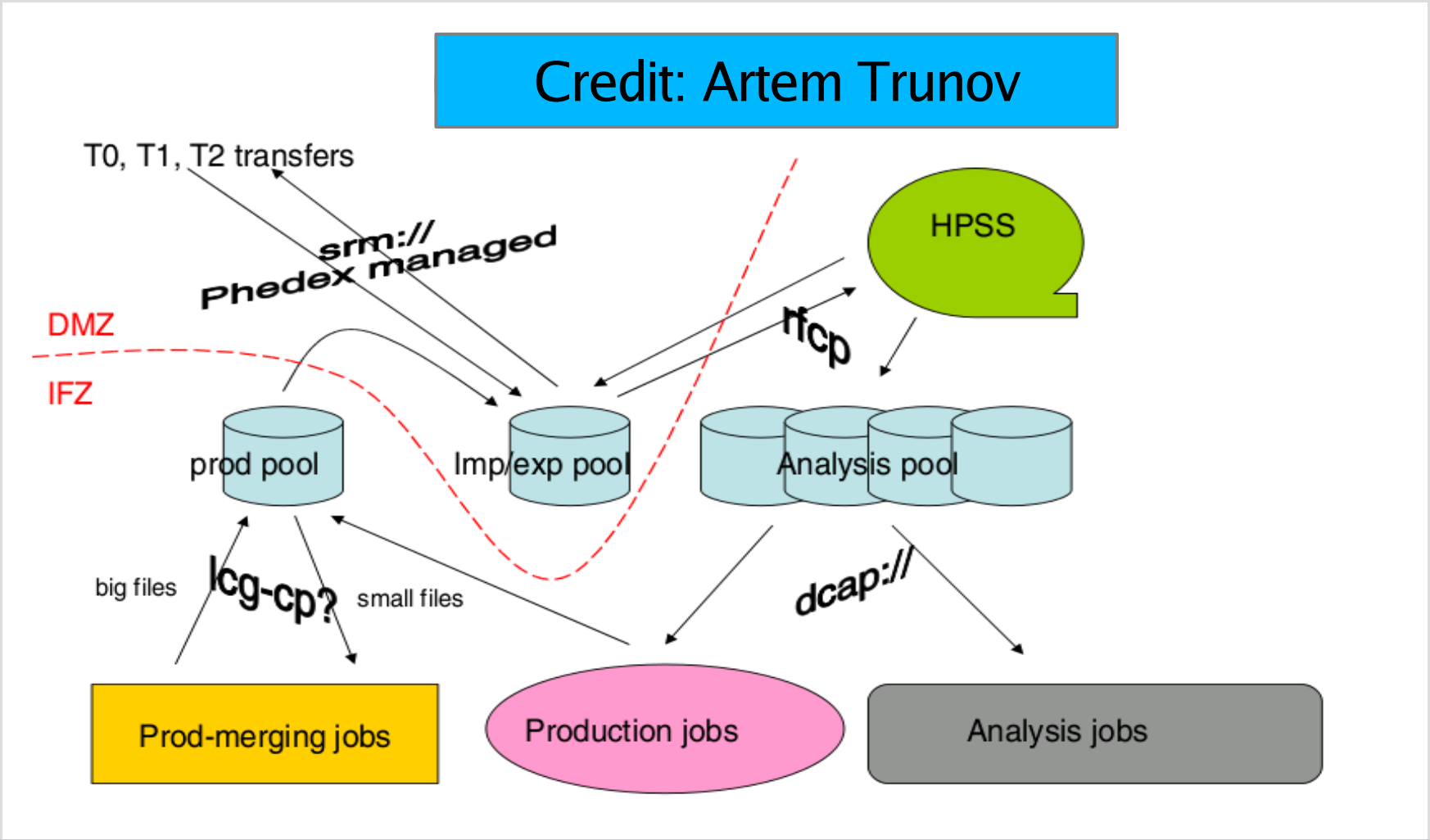
Separate resources

- Storage Groups used to:
  - Separate resources for a better usage
  - Separate data for a better performance
  - Make the maintaining easier
- dCache input parameters determine a storage group
  - Client IP or subnet
  - I/O operation (read, write, cache)
  - Path of the file (in the dCache namespace)
- dCache allows multi-queues used for different protocols (GFTP, dCap)

Read vs Write pools

- Write pools receive files from other sites with a high (but very well known) throughput
  - Disk access tuned for sequential write
  - Network tuned for high throughput
  - High quality hardware required for precious files (but small volume needed)
- Read pool are used by local jobs
  - Disk access tuned for random access by lots of jobs
  - High quality is not required (files can be retrieved from HPSS to other pools) but large volumes are needed
- RW pools are also used for traffic in both directions

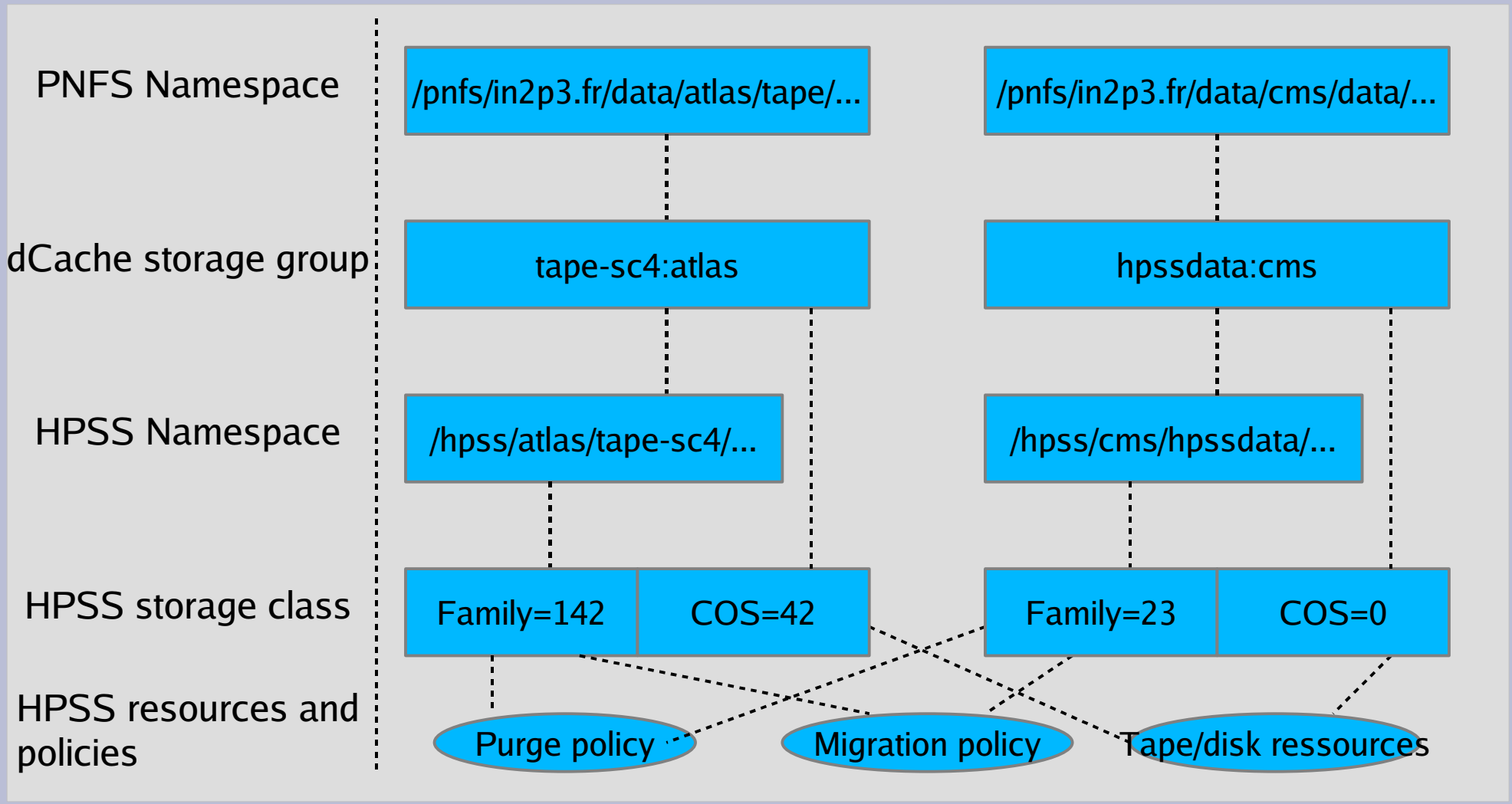
CMS dCache pools setup



dCache/HPSS SC mapping

- HPSS Classes Of Service are defined in dCache namespace tags (PNFS hidden files)
  - Then used by migration script
- HPSS Families are used in dCache through the mapping between the dCache storage group and the HPSS namespace (see example on next slide)

Storage Classes levels



SRM v2.2 Storage Classes

- The following slides are not a description of how we will do it but rather a proposition on how we could do it opened to discussion
- Storage classes implementation should be an agreement between the site's requirements and the experiment's requirements
- As they are roughly the same (have a performant and optimised system), this should not be difficult
- As usual, IN2P3 T1 works closely with experiment representatives

Implementation

- Tape1Disk0
  - Probably better to separate buffer for incoming data to buffer for outgoing data
  - Even separate buffer for RAW data to buffer for ESD...
  - Seems not an issue (throughput known, source/dest known)
- Tape0Disk1
  - Strong collaboration between site and experiment representative to size the pool, the number of servers...
  - Probably separate production pools from user pools to avoid queues bottleneck because user intercations are unpredictable

Implementation (2)

- Tape1Disk1
  - From the experiments' requirement, it seems Tape1Disk1 could be implemented by automatically moving files from transfer pools to disk pools (dCache “file hopping”)
  - Some more discussions needed with experiments representatives

Specific use cases

- Atlas requires ESD (and AOD) on T1D0 at production site and on T0D1 at production site and paired T1
  - Duplicate files?
  - Use T1D1 instead?
  - Use dCache “file hopping” (automatically moves files from transfer pools to disk pools)?
- LHCb: RAW (T0D1) must stay on disk 48h
  - Cannot assure with T0D1 system-managed class
  - Use T1D1?
  - Pin for 2 days?