An update on SRM

- Mumbai & subsequent discussion summary
  - http://agenda.cern.ch/fullAgenda.php?ida=a056461
    - See the Storage Types BOF at the end of the agenda, with the material attached.
  - Workshop summary (version 2):
    - http://litmaath.home.cern.ch/litmaath/Mumbai-SC4.pdf

- v2.x shopping list

- More...

Mumbai SC4 in a nutshell

- Tony Cass: “we are the masters and can decide what words mean in our environment. This we did in Mumbai: permanent means with tape copy, durable means without. ”

- Permanent
  - System manages cache
  - Access sometimes slow

- Durable
  - User (VO) manages cache
  - Access fast

How to implement it

- Let info system publish an SAPath + SAType per storage class
  - Path can be the same or different for both classes
- Modify clients to pick up the correct path as instructed by user
  - Backward compatible default behavior
- Client shall set “wantPermanent” flag corresponding to SAType
  - True → permanent, false → durable, null → server decides

Example for dCache

dn: GlueSALocalID=dteam-durable,GlueSEUniqueID=srm.my_domain,mds-vo-name=…
[…]
GlueSARoot: dteam:/pnfs/my_domain/foo/dteam
GlueSAPath: /pnfs/my_domain/foo/dteam
GlueSAType: durable
[…]
GlueChunkKey: GlueSEUniqueID=srm.my_domain
[…]

dn: GlueSALocalID=dteam-permanent,GlueSEUniqueID=srm.my_domain,mds-vo-name=…
[…]
GlueSARoot: dteam:/pnfs/my_domain/bar/dteam
GlueSAPath: /pnfs/my_domain/bar/dteam
GlueSAType: permanent
[…]
GlueChunkKey: GlueSEUniqueID=srm.my_domain
[…]

Example for CASTOR

dn: GlueSALocalID=dteam-durable,GlueSEUniqueID=srm.my_domain,mds-vo-name=...
[...]
GlueSARoot: dteam:/castor/my_domain/foobar/dteam
GlueSAPath: /castor/my_domain/foobar/dteam
GlueSAType: durable
[...]
GlueChunkKey: GlueSEUniqueID=srm.my_domain
[...]

dn: GlueSALocalID=dteam-permanent,GlueSEUniqueID=srm.my_domain,mds-vo-name=...
[...]
GlueSARoot: dteam:/castor/my_domain/foobar/dteam
GlueSAPath: /castor/my_domain/foobar/dteam
GlueSAType: permanent
[...]
GlueChunkKey: GlueSEUniqueID=srm.my_domain
[...]

Example usage

- Lcg-utils/GFAL

$ lcg-cr --stype durable file:... -d castorsrm.cern.ch -l lfn:...

Default:

$ lcg-cr --stype permanent file:... -d castorsrm.cern.ch -l lfn:...

For SRM v2.x it might become "--stype durability:accessibility:lifetime"

- FTS
  - Option to set storage class per transfer
  - VO plug-in can determine correct SURL
    - Could cache results of BDII query to find all SAPaths
    - Default could use "generated" directory under correct SAPath

Graeme Stewart's matrix

Storage Type Requested	Storage Types Found	Action
permanent	permanent	copy to permanent path
permanent	durable	error
durable	permanent	error
durable	durable	copy to durable path
null	permanent	copy to permanent path
null	durable	copy to durable path
null	both	copy to permanent path

- Mixed client/server responsibility
- Some servers may not recognize the indicated error conditions

Don Petravick's Storage Ontology

Abstract concept	SRM v1 web svc API notes	SRM v2 web svc API notes	GLUE notes
File retention time	not modeled	permanent, durable, volatile	permanent, durable, volatile, other (SAType)
Quality of retention	not modeled	not modeled	tape, (multi)disk, other (implicit in SE architecture)
Uniformity of transfer performance	not discoverable, manageable via pin	not discoverable, manageable via prepareToGet (with side effects)	tape, (multi)disk, other (implicit in SE architecture)

Experiment's view

- LHCb and others distinguish these classes:
  - Files that are rarely used but of high value
    - High quality of retention → tape
  - Files that are frequently used, less costly to regenerate or retrieve from other replica
    - Disk (precious files also on tape)
  - Files that are frequently used, are highly mutable, often small, but need fast access and resilience
    - User files → disk and tape (unless scratch)
- LHCb want to put storage type in path
  - Can be done in VO namespace, not a requirement on SE implementation

Tier-1 view

- Lionel Schwarz (IN2P3 Lyon):
  - Big sites may need multiple SAPaths per class!
    - The clients had better deal with that

- Jos van Wezel (FZK):
  - “At the moment no one is charging real money but the moment you do that users must have the ability to choose the storage based on price/speed/reliability etc.”
    - May need to extend the ontology even further

Critical features for WLCG

- Result of WLCG Baseline Services Working Group
  - http://cern.ch/lcg/PEB/BS
- Originally planned to be implemented by WLCG Service Challenge 4
  - Delayed until autumn 2006
- Features from version 1.1 + critical subset of version 2.1

(Nick Brook, SC3 planning meeting – June '05)

  - File types
  - Space reservation
  - Permission functions
  - Directory functions
  - Data transfer control functions
  - Relative paths
  - Query supported protocols

File types

- Volatile
  - Temporary and sharable copy of an MSS resident file
  - If not pinned it can be removed by the garbage collector as space is needed (typically according to LRU policy)
- Durable
  - File can only be removed if the system has copied it to an archive
- Permanent
  - System cannot remove file
- Users can always explicitly delete files
- For SC4 the experiments only want durable and permanent

Space reservation

- v1.1
  - Space reservation done on file-by-file basis
  - User does not know in advance if SE will be able to store all files in multi-file request
- v2.1
  - Allows for a user to reserve space
    - But can 100 GB be used by a single 100 GB file or by 100 files of 1 GB each?
    - MSS space vs. disk cache space
  - Reservation has a lifetime
  - “PrepareToGet(Put)” requests fail if not enough space
- v3.0
  - Allows for “streaming”
    - When space is exhausted requests wait until space is released
  - Not needed for SC4
- What about quotas?
  - Strong interest from LHC VOs, but not yet accepted as task for SRM

Permission functions

- v2.1 allows for POSIX-like ACLs
  - Can be associated per directory and per file
  - Parent directory ACLs inherited by default
  - Can no longer let a simple UNIX file system deal with all the permissions
    - Need file system with ACLs or ACL-aware permission manager in SRM
      - May conflict with legacy applications
- LHC VOs desire storage system to respect permissions based on VOMS roles and groups
  - Currently only supported by DPM
- File ownership by individual users not needed in SC4
  - Systems shall distinguish production managers from unprivileged users
    - Write access to precious directories, dedicated stager pools
    - Supported by all implementations

Directory functions

- Create/remove directories
- Delete files
  - v1.1 only has an “advisory” delete
    - Interpreted differently by different implementations
      - Complicates applications like the File Transfer Service
- Rename directories or files (on the same SE)
- List files and directories
  - Output will be truncated to implementation-dependent maximum size
    - Full (recursive) listing could tie up or complicate server (and client)
      - May return huge result
      - Could return chunks with cookies → server would need to be stateful
    - It is advisable to avoid very large directories
- No need for “mv” between SEs

Data transfer control functions

- StageIn, stageOut type functionality
  - prepareToGet, prepareToPut
- Pinning and unpinning files
  - Avoid untimely cleanup by garbage collector
  - Pin has a lifetime, but can be renewed by client
    - Avoid dependence on client to clean up
- Monitor status of request
  - How many files ready
  - How many files in progress
  - How many files left to process
- Suspend/resume request
  - Not needed for SC4
- Abort request

Relative paths

- Everything should be defined with respect to the VO base directory
- Example:

  srm://castorsrm.cern.ch/castor/cern.ch/grid/lhcb/DC04/prod0705/0705_123.dst

- SE defined by protocol and hostname
- VO base directory is the storage root for the VO
  - Advertized in information system, but unnecessary detail
    - Clutters catalog entries
    - SRM could insert VO base path automatically
      - Available in dCache
- VO namespace below base directory

Query supported protocols

- List of transfer protocols per SE available from information system
  - Workaround, complicates client
  - SRM knows what it supports, can inform client
- Client always sends SRM a list of acceptable protocols
  - gsiftp, (gsi)dcap, rfio, xrootd, root, ...
  - SRM returns TURL with protocol applicable to site
- Query not needed for SC4

More coordination items

- SRM compatibility tests
  - Test suite of Jiri Mencak (RAL) seems to do the job for us
- Clients need to keep supporting v1.1
  - First try v2.x?
- v2.x on separate port
  - 8444 standard?

- xrootd and rootd integration
- rfio incompatibility
- Quotas for user files
- ...