SRM v2.2 planning

- Critical features for WLCG
- Results of the May 22-23 workshop at FNAL
  https://srm.fnal.gov/twiki/bin/view/WorkshopsAndConferences/GridStorageInterfacesWorkshop
- SRM v2.2 definition geared to WLCG usage, but still compatible with other implementations
- Some notions backported from SRM v3, others added for WLCG
- WLCG “MoU”
  https://srm.fnal.gov/twiki/pub/WorkshopsAndConferences/GridStorageInterfacesWSAgenda/SRMLCG-MoU-day2.doc
  - Needs some updates and polishing
- Schedule for implementation and testing
  https://srm.fnal.gov/twiki/pub/WorkshopsAndConferences/GridStorageInterfacesWSAgenda/Schedule.pdf
- Friday phone conferences to monitor progress and discuss issues

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
- The experiments only want to store files as permanent
  - Even scratch files → will be explicitly removed by experiment

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 etc.
      - 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 files or directories (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/offsets → server might 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
- (a way for) 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://srm.cern.ch/castor/cern.ch/grid/lhcb/DC04/prod0705/0705_123.dst

- SE defined by protocol and hostname (and port)
- VO base directory is the storage root for the VO
  - Advertized in information system, but unnecessary detail
    - Requires information system lookup for storing files
    - Clutters catalog entries afterwards
    - 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, http(s), ...
  - 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)
  - Test suite for GGF-GIN by Alex Sim (LBNL)
  - Test suite of Gilbert Grosdidier (LCG)
  - ...
  - Which one(s) will do the job for WLCG?
- Clients need to keep supporting v1.1
  - First try v2.x?
- Some implementations need v2.x to be on separate port
  - 8444 standard?
- xrootd integration
- rfio incompatibility
- Quotas for user files
- ...

SRM v2.2 MoU for WLCG

- Summarize agreed client usage and server behavior for the SRM v2.2 implementations used by WLCG applications
  - Servers can ignore non-WLCG use cases for the time being
- Clients
  - FTS, GFAL, lcg-utils
- Servers
  - CASTOR, dCache, DPM

Storage classes

- Stick with SRM v3 terminology for now, but with a WLCG understanding

- TRetentionPolicy {REPLICA, CUSTODIAL}
  - OUTPUT is not used
- TAccessLatency {ONLINE, NEARLINE}
  - OFFLINE is not used

- Tape1Disk0 == CUSTODIAL + NEARLINE
- Tape1Disk1 == CUSTODIAL + ONLINE
- Tape0Disk1 == REPLICA + ONLINE

- All WLCG files (SURLs) are permanent
  - Files can only be removed by the user

Information discovery

- WLCG does not need an SRM information interface for the time being
  - Client implementations provide list of required information
  - GLUE schema will be modified accordingly

- An interface to obtain (all) the relevant information can be defined later
  - Would allow the SRM clients and servers to be self-sufficient
  - Would simplify the information provider implementations

srmReserveSpace

- Only deals with disk
  - Cache in front of tape back-end, and disk without tape back-end
  - Tape space considered infinite
- TapeNDiskM storage classes only require static reservations by VO admins
  - Can be arranged out of band without using the SRM interface (CASTOR)
    - Agreement between VO admin and SE admin will be needed anyway
  - Networks of main clients can be indicated (dCache)
- Dynamic reservations by ordinary users not needed in the short term
  - At least CMS want this feature in the medium term
- userSpaceTokenDescription attaches meaning to opaque space token
  - “LHCbESD” etc.

srmLs

- To get all metadata attributes for individual files, but only some for directories
  - Directory listings quickly become very expensive
- Directory listing use case would be to check consistency with file catalog
  - An implementation-dependent upper limit will apply for the time being
    - Use of the offset and count parameters requires further discussion
- TFileLocality {ONLINE, NEARLINE, ONLINE_AND_NEARLINE, LOST, NONE, UNAVAILABLE}

srmPrepareToPut

- To store a file in the space (i.e. storage class) indicated
  - WLCG clients will supply the space token
- WLCG files are immutable, cannot be overwritten
- TConnectionType { WAN, LAN }
  - Will be set by FTS (for 3rd party transfers)
- TAccessPattern { TransferMode, ProcessingMode }
  - ProcessingMode would apply to a file opened by GFAL (but not via lcg-utils)

srmPrepareToGet

- To prepare a file for “immediate” transfer or access
  - Recall from tape and/or copy to pool accessible by the client should now be done through srmBringOnline
- WLCG usage excludes changing space or retention attributes of the file
- TConnectionType { WAN, LAN }
  - Will be set by FTS (for 3rd party transfers)
- TAccessPattern { TransferMode, ProcessingMode }
  - ProcessingMode would apply to a file opened by GFAL (but not via lcg-utils)

srmBringOnline

- To indicate that a prepareToGet for the files is expected in the near future
  - A delay parameter can be used for further optimization
  - A prepareToGet could tie up resources, e.g. I/O movers in dCache
- Signature very similar to that of prepareToGet
  - No TURLs are returned

srmCopy

- To copy files or directories between SEs
  - Directories will not be supported for the time being
- srmPrepareToGet and srmPrepareToPut restrictions apply
- Individual copies in a multi-file request can be aborted
  - Target SURLs uniquely identify the copy requests
- removeSourceFiles flag has been deleted from the specification
  - Too dangerous...

srmChangeSpaceForFiles

- To change the storage class of the given files
  - Tape1Disk0 ←→ Tape1Disk1 (add/remove disk copy)
  - Tape0Disk1 ←→ Tape1DiskN (add/remove tape copy)
- To be decided which transitions shall be supported
- The SURL shall not be changed
  - Absolute path may change if SURL only contains relative path (as desired)
- Not required in the short term

Removal functions

- srmRm
  - Remove SURL
- srmReleaseFiles
  - Removes pins e.g. originating from prepareToGet
  - May flag disk copies (TURLs) for immediate garbage collection
- srmPurgeFromSpace
  - As previous, but not associated with a request
- srmAbortFiles
  - To abort individual copy requests
- srmRemoveFiles has been deleted from the specification

Schedule (1/3)

- WSDL and SRM v2.2 spec - June 6
  - Various inconsistencies have been fixed since
  - Discussion about the need for some unexpected changes w.r.t. v2.1
  - Still to be examined by Timur for dCache

- srmPrepareToGet, srmPrepareToPut at the same level of functionality as it is present now - June 20
  - Not technically challenging
  - Need 3 endpoints by the end of this period
  - Need a test suite, Java, C and C++ clients are included
    - LBNL tester
    - FNAL srmcp - Apache Axis + Globus CoG Kit
    - Castor C++ client – gSoap + GSI plugin
    - DPM C client – gSoap + GSI plugin

Schedule (2/3)

- Compatibility 1 week after that - June 27
  - ML to run the tests and work with the developers
- dCache srmCopy compatibility with DPM and Castor srmPrepareTo(Get/Put) - work by Fermilab - July 4
- Space Reservation prerelease implementations - Sept 1
  - To coincide with SRM v2/v3 workshop at CERN, Aug 30 – Sept 1
- Space Reservation / Storage Classes - Sept 30 (optimistic)
  - Proper SRM or out-of-band way to reserve space
  - srmGetSpaceTokens
  - Modifications to srmPrepareToPut and srmCopy; srmPrepateToGet optional
  - srmRm, srmReleaseFiles (srmPurgeFromSpace not needed)
- Space Reservation may only work for special deployment configurations
  - Need to determine (per VO) if disk pools should be externally reachable

Schedule (3/3)

- srmBringOnline - Oct 6
- srmLs - return of space tokens is not required for October
- WLCG clients should follow the same schedule
  - Ready to be used as testers by the end of Sept
  - Will have several SRM test suites
  - Functionality, stress tests, error handling and resilience to “malicious” clients
- Integration week at RAL - Oct 9-13
  - Firm dates to be decided as milestone (by end of June)
- It could all work sufficiently by Nov 1
  - To allow v2.2 to become the standard SRM service (v1.1 for legacy apps)
  - Development of less urgent features will continue