SRM 2.1 working group update

- Phone conferences monthly → weekly at Fri. 16:00 CEST
- FNAL workshop May 22-23

- New ontology documents prepared and discussed
  - Tony
    - Axes of SRM properties/qualities
  - JPB + James
    - SRM Storage and File Types (v4)
      - http://litmaath.home.cern.ch/litmaath/MB/SRM_Storage_and_File_Types-v4.pdf
    - Work back from SRMv3 as much as possible
  - Olof
    - Cache attributes

Durable vs. permanent

- Volatile/durable/permanent are about the lifetime
  - PUT: namespace
  - GET: cache

- Durable type (as defined in SRMv2/v3) considered not useful for WLCG
  - alerting admin when file lifetime expires is unworkable
  - experiments only want permanent files
    - volatile files for scratch are not needed either, as experiments do their own bookkeeping
  - argument for durable files: they do not use up tape quota
    - “do not send these to tape yet, they must still be validated”
    - better implemented by supplying cache attributes on SRM PUT

- Custodial responsibility: technical choices must be advertized
  - user can choose out of what is available
  - enumerate the possible STORAGE CLASSES (term agreed during meeting)

Storage classes

	min. required copies		Mumbai term
	Tape	Disk	
Storage Class			
A	1	0	"permanent"
B	1	1	"permanent-durable"
C	0	1	"durable"
D	0	>1	
E	>1	0	
F	>1	1	
G	>1	>1	
...	...	...	

- Instead of A/B/C/… the names would rather be srmTape1Disk0 etc.

PUT vs. GET

- PUT
  - Add storage class argument
  - Also keep storage type argument, because other users may need it
    - P/D/V only indicates expiration time
  - New method needed to change a file's storage class
    - Only for privileged users

- GET
  - Not symmetric to PUT
  - Class A would need volatile type → system managed cache
  - Class B/C would need permanent → user managed cache
    - But the permanent copy may be in the wrong pool (e.g. LAN vs. WAN)
    - A volatile copy can still make sense
  - Extra cache attribute parameters to indicate intended usage
    - LAN vs. WAN
    - Random vs. sequential
    - ...

Cache attributes

- LHCb: LAN access via rfio/dcap/root, WAN access via gridftp
- Alice: rfcp all data to and from WN
  - expensive
  - need to be directed to pool with adequate parameters
- Atlas: low-rate gridftp access from T2
  - gridftp over WAN need not always be fast (even on the OPN)

- Transfer speed to be matched to pool parameters
  - do not want high-speed transfer slots used for a slow site
  - do not want a low-rate pool allowing many concurrent connections to be hit by high-rate transfers

- In the end about 4-5 access patterns to be mapped

New methods

- Timur
  - “bringOnline” function separate from prepareToGet
    - Latter starts an I/O server in dCache
- Olof
  - Asynchronous prestage function w/o request token
    - But then it cannot be canceled
  - Asynchronous space reservation
    - Need to control fragmentation
- JPB
  - prepareToGet == bringOnline == prestage
  - I/O server can be started on open or statusOfGetRequest

GLUE considerations

- Need query/ls functions to advertize and find out what is available
- First agree on necessary SRM functionality, then adapt schema as needed
  - Storage classes
  - Cache attributes
- Use schema extensions where possible?
  - A new minor version probably cannot be avoided
- A lot is not used today
  - Drop or fix?
- What does free space mean?
  - Cache or back-end?
  - What if there are multiple SARoots?
- Changes may be driven by FTS/GFAL/lcg-util/… examples