

GGF 10 GSM  
11 March, 2004  
Berlin, Germany

Grid Storage Management Working Group  
Chairs: Arie Shoshani (LBNL) Peter Kunszt (CERN)  
Secretary: Alex Sim (LBNL)

- Working group web : <http://sdm.lbl.gov/gsm>
- GSM Charter Session Agenda
- Charter : GSM Focus and purpose  
Standardization effort to define a Storage Resource Management SRM interface  
Storage Resource Managers are
  - Middleware components whose function is to provide dynamic space allocation and file management of shared storage components on the grid
  - They complement Compute Resource managers and Network resource Managers
    - C: What is the Storage Resource and what is SRM managing in addition to above two? It could include such as network resource... pseudo gridftp server, load balancing...
- Where do SRMs belong in the Grid architecture / the grid picture
- SRMs support a brokering service by invoking transfer services: DataMover as brokering part....
- Charter : GSM scope
  - Focus on the definition of the functionality of s std SRM interface
  - A short term effort to have a workable interface that grid projects could immediately make use of to resolve interoperability issues between storage interfaces
  - We solicit active participation from all interested parties
  - SRM has already a strong user/provider community
    - In our experience so far the SRM interface was one of the most successful demonstrators of the interoperability effort in diff areas in US and Europe
- History
  - 3 years of SRM management activity
  - Q: In GGF7 what was there?  
A: We had discussions on the charter and see if it was a appropriate.
- Participants in SRM efforts so far
- GSM goals:
  - Produce a GGF recommendation documents for SRM interfaces
    - Based on the work that has been done to date
  - Description of an agreed certification test suite
    - Certify if an implementation truly compiles with the given interface recommendation
  - Aggressive timeline for documents

- Previous work on SRM available today
  - Draft GSM spec to follow this year
- SRM functionality list
  - Manage spaces
  - Manage files (in spaces)
  - Manage directories
  - Manage multi-file requests
  - Access remote sites for files
  - Accounting
  - Access control
  - Q: about data access? A: we will talk about this later.
- SRM functionality details
  - Manage spaces dynamically
    - Reservation, lifetime
    - Manage multiple spaces per client
    - Negotiation (coordinate with GRAAP)
    - Types of spaces: shared, user owned
  - Manage files in spaces
    - Request to put files in spaces
    - Request to get files into spaces
    - Manage default spaces
    - Lifetime, pinning of files, release of files
    - No logical name space management (rely on GFS)
  - Manage directory structures
    - Usual unix semantics
    - srmLS, srmMkdir, srmMv, srmRm, srmRmdir
    - a single directory for all spaces
    - file assignment to spaces is virtual
    - Manage multi-file requests
      - manage request queues
      - manage caches
      - Q: why not putting this item in the file management section, but in the directory structures section?  
A: You're right. Cache should be managed for one file as well. Here is the interesting place because of the queue.
      - Q: Caches are different from the reserved space?  
A: Cache has nothing to do with the interface. This is a consequence of the functionality, but not for the interface. Space is better term than cache.
      - Q: why not doing with the ftp?  
A: There is an additional rich meta data. Storage should have gridftp as well.  
C: Gridftp is for moving the data. We also want to talk to one interface.
      - Q: Name space manipulation should be outside of this WG?

A: in the short term, well, no choice. In SRM, there is transfer protocol: http, ftp, etc. If you want SRM to negotiate protocols, we need this. Two protocols can make a conflict behavioral result: one uses and the other removes the file.

- C: Storage management, the term in industry is RAW storage. It is little bit confusing from the industrial perspective.

A: we intend to sit on top of the raw. You're right. This is not the same storage.

- Q: Why not database storage?

A: Granularity of the community is file. Why only files? Because it's easy to deal with. It could be an extension of this... what we're trying to avoid is the content of the file. If DB, it's a chunk of the data. Space management concept is probably an extension of this.

- C: We could have middleware storage, rather than just storage...

A: This is like a storage broker... then? Yes, it can combine 10 NAS into one virtualized.. it goes beyond allocating... - it's not really virtual... its underlying something that does the virtualization. This is not a file system concept. The answer is neither. Broker is to go out and find someone to do. But SRM does the job.

- A: There is no name space management in SRM. It relies on some one else's work. It does not duplicate much work from others.

- Q: SAN std is at the bottom, how does SRM std match the underlying std?

A: Depending on the bottom layer, the layer SRM can be thin and/or thick. We'll see.

- Q: is this an effort of yet another problem?

A: There is no product. This is only an interface definition. End product is the interface. SRB can have an interface of SRM. Not the other way around.

- C from Peter Clark: This is not a piece of software. Purpose of having this group is to have a useful interface, by definition.

- access remote sites for files

- bring files from other sites and SRMs as requested
- use other services: transport (gridftp), maybe catalog registration (OREP)

- Q: can you specify what kind of spaces in the request?

A: yes.

- Q: how can you enforce the usage?

A: we have accounting...

C: It should be in the storage management part, not in the namespace management part.

- Accounting:
  - Keep track of space usage
  - Mbytes-hours (bytes-seconds)
  - Capability-based quota allocation assigned by a VO
  - Interfaces to enable usage reporting
- Access control
  - Manage ACLs
  - Synch ACLs in replicas is an option problem
  - Master copy concept maybe necessary to enforce access control
- SRM functionality list : v1.x + v2.x
  - Comparisons of two versions
- Relationship to DAIS
  - What is “file access”
    - File movement – SRM focus
      - Get the file into my space from storage system
      - T a file from my space into storage system
      - The file is a “big” of bits.
    - Look into the content of the file – DAIS focus (maybe)
      - Equiv. to queries to a db system
        - E.g. SQL, XQuery
      - Gateway to data systems
      - Structure of files has to be exposed
- Phases of document specifications
  - Phase 1: manage spaces, manage files (in spaces), manage dirs, manage multi-file requests – only managing local resources
    - Q: queuing is not exposed in the API.  
A: we could introduce priorities in the interface.
  - Phase 2: access remote sites for files
  - Phase 3: accounting, access control
  - C: Exception handling and etc... has to be specified in the spec.
- GSM deliverables and milestones
  - C from Ian bird: phase 2 should be based on the phase 1. Suggesting to have an evaluation period between phases.
  - C: Have an evaluation period between phases.
  - Q (Peter C): is SRM v1 phase 1 or 2 ?  
A: No. We would have GGF version 1. People not in the community could participate. Learning from the experience so far, we would have the first GGF version 1.
- Discussion of GSM management and processes
  - Definition and rules of active and passive participation
    - Active – real commitment
      - Core group writing doc and defining specs
      - People how implement SRMs

- People who implement clients to SRMs
  - Current core group exist – we solicit new members
- Passive – real interest
  - Mailing list by sign up
  - Read and comment doc to core group coordinator (Arie) – we solicit opinions on our documents
- How to resolve conflicts so that we can stay in the timeline
  - Only core group members by vote
  - Details of voting weight to be worked out inside core group
- C: working group has to have a consensus.
  - Core group = design team

○