SRM: Expt Reqts

Nick Brook

- Revisit baseline services working group
- Priorities & timescales
- Experiences & Use case

"Telling Grandma how to suck eggs"

SRM is more than a way to copy files to & from SE - more than a transfer method

SRM interface needed for all SE - LCG-DPM, dCache, CASTOR, DRM …

Baseline Services report

- All experiments require SRM at all sites
- The WG has agreed a common "LCG-SRM" set of functions, that the experiments need: (CMS ratification missing)
  - SC3: v1.1
  - SC4: LCG-SRM
- LCG SRM functionality:
  - V1.1 + space management, pin/unpin, etc
  - Not full set of V2.1
  - V3 not required
- Coordination group with SRM developers set up in April workshop
  - Slowed down/stopped?
  - This talk - an attempt to kick-start (fault lies with expts)
- Most apps will use ROOT (via POOL or direct) to access data
  - ROOT will interface to SRM

Basic SRM functions

(see link from baseline services group web pages - http://cern.ch/lcg/PEB/BS)

File types:

- Volatile - temporary & sharable copy of a MSS resident file - if not pinned can be removed by garbage collector
- Durable - file cannot be removed automatically. If space needed file may be copied to MSS …
- Permanent - system cannot remove file

Expts only require - volatile & permanent file types

Basic SRM functions

Space reservation:

SRM v1.1: space reservation done on file-by-file basis

- User doesn't know in advance if SE will be able to store all files in request

SRM v2.1: allows for a user to reserve space

- Reservation has a lifetime
- Data "PrepareToGet(Put)" requests fail if not enough space

SRM v3.0: allows for "streaming"

- When space is exhausted new requests don't fail but wait until space is released

Expt happy with v2.1 space reservation functionality

Basic SRM functions

Permission functions:

SRM v2.1: allows for a posix-like ACLs

- Can be associated with each directory or file

Expts desire storage system to respect permissions based on VOMS roles & groups

Expt have NO wish for file ownership by individual users

Basic SRM functions

Directory functions:

- Create/remove directories
- Delete files
- Rename directories or files (on a particular SE)
- Directory listing (not necessarily recursive listing)
- No need for "mv" (between SRM SE's)

Basic SRM functions

Data transfer functions (misnomer - not actual data movement but to prepare access to data):

- stageIn, stageOut type functionality
- Pinning & unpinning functionality
- Request token to monitor status of request
  - How many files ready
  - How many files in progress
  - How many files left to process
  - Suspend/re-start/abort request

Basic SRM functions

Relative paths:

Everything should be defined with respect to the VO base directory

- srm://castorsrm.cern.ch/castor/cern.ch/grid/lhcb/DC04/prod0705/0705_123.dst
  - Define SE
  - Site definition for VO
  - VO definition

{ defines VO SE

Basic SRM functions

Query the protocols supported by site/SE:

- Function already in information system
- List of protocols supported by VO can be given in the application to SRM - return TURL with protocol applicable to site

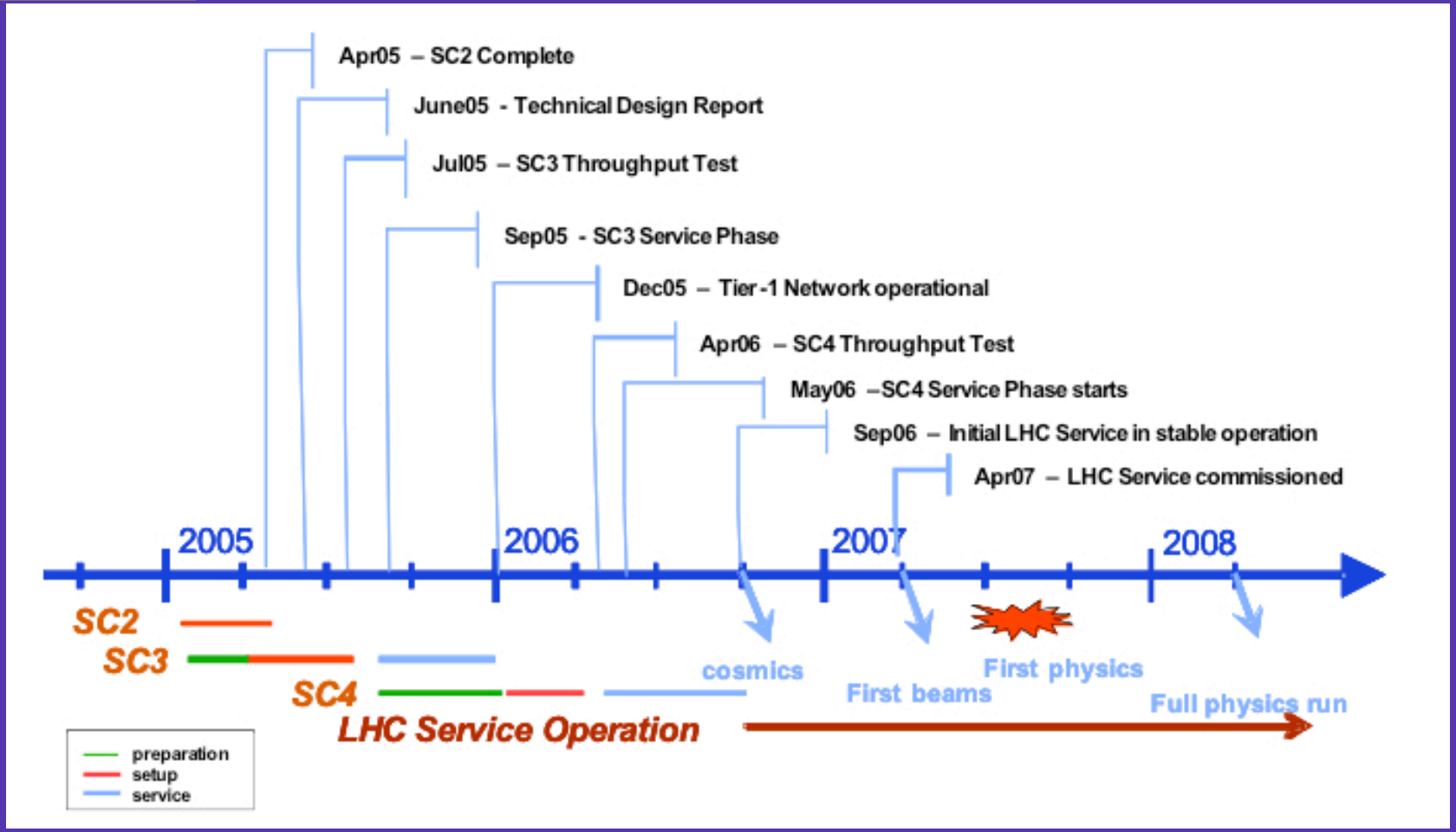
Prioritised List

In descending order:

1.Pin/Unpin functionality
2.Relative paths in SURLs
3.Permission functions: All experiments would like the permissions to be based on roles & DN's. SRM should be integrated with VOMS
4.Directory functions (with the exception of "mv")
5.Global space reservation: ReserveSpace, ReleaseSpace and UpdateSpace, though CompactSpace is not needed
6.srmGetProtocols is seen as useful but not mandatory
7.AbortRequest, SuspendRequest and resumeRequest not seen as necessary

First five seen as essential

Timescales



SC4 (Feb'2006) - "to have available all missing tools & existing components"

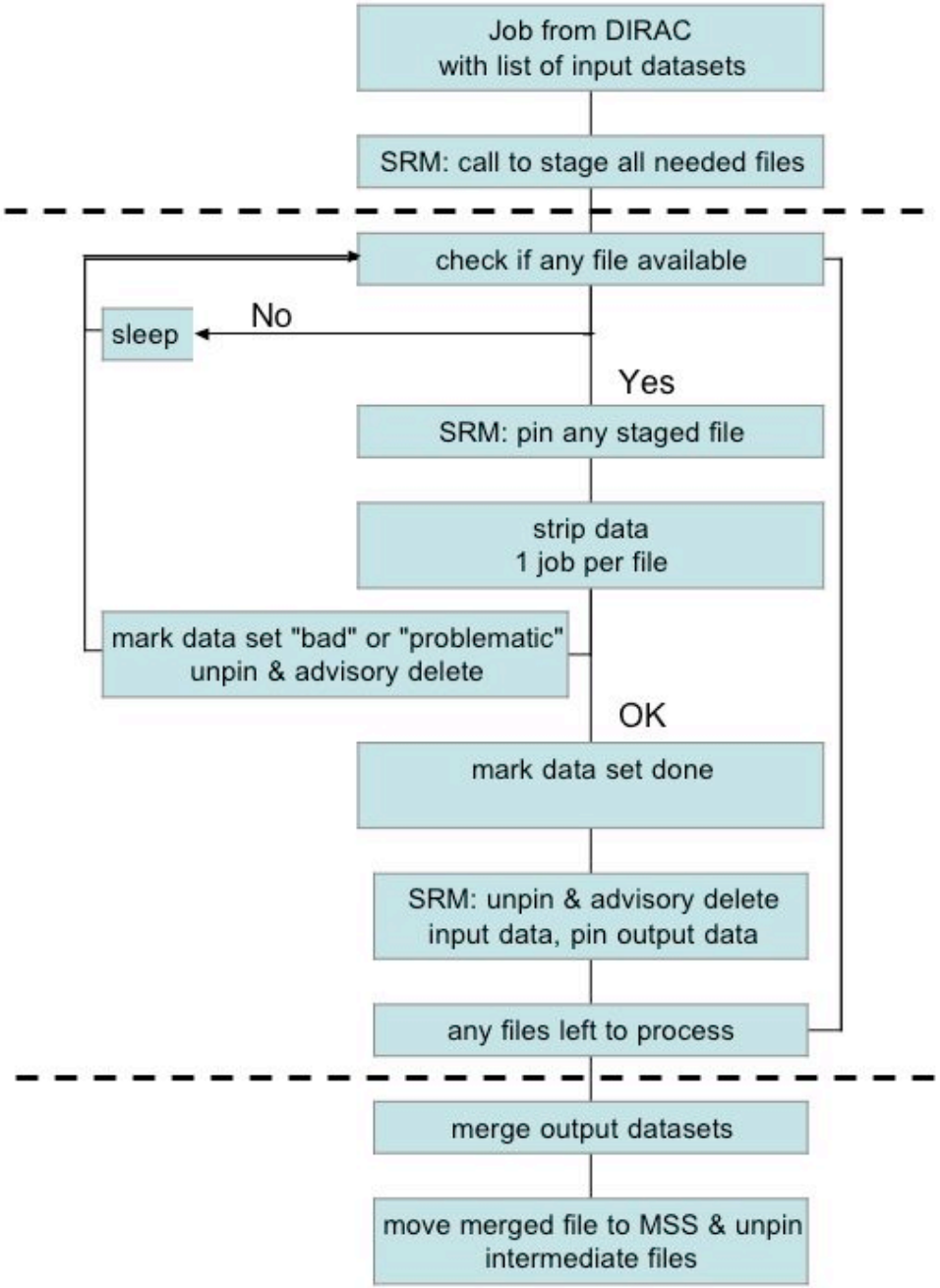
Necessary to expose service well before start of SC4

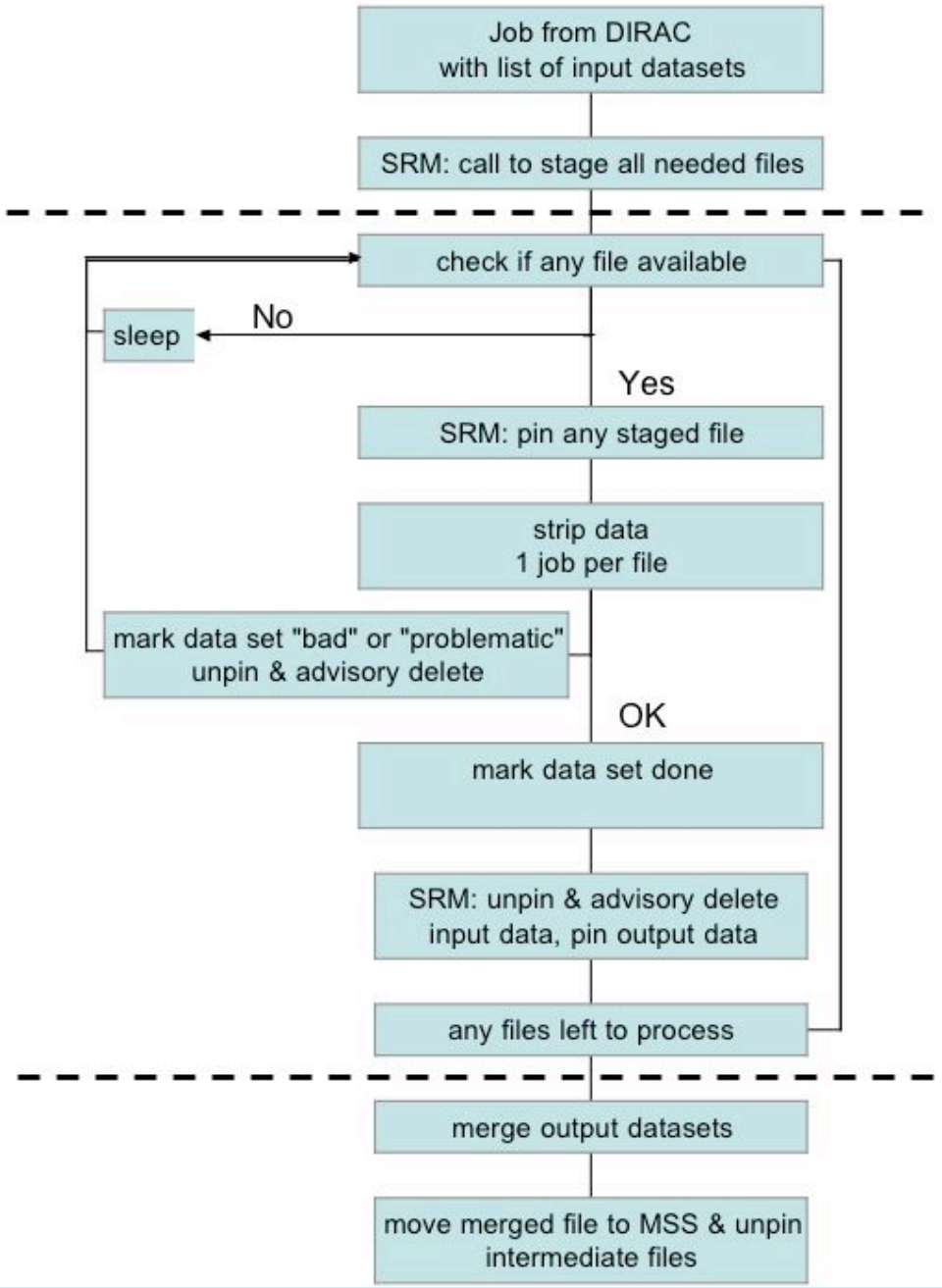
Use case

LHCb

Stripping:

- centralised analysis
- Reduced reconstructed dataset about factor of ~10
- Performed 4 times a year: twice with recons & two other times
- Need to retrieve RAW & rDST data from MSS
- Output files distributed to all Tier-1 centres





Current Experience & Usage

- Jobs have several input files (between 40 and 80)
- Jobs sent to site where the data are placed
- Currently 3 sites used CNAF, CERN and PIC based on CASTOR Mass Storage
- Using SRM interface to access MSS

Scale of stripping in Data Challenge

Physics stripping jobs	
Number of events per job	40,000
Number of files	80
Input data size	80*0.3 = 24 GB
Number of output files	2 (DST + event collection)
Output DST size	600 MB
Event collection size	1.2 MB
Number of events	60M
Number of jobs	1,500
Input data size	36 TB
Output data size	0.9 TB
Trigger stripping jobs	
Number of events per job	360,000
Number of files	400 (files of 900 evts) or 200 (1800 evts)
Input data size	400*0.18 = 72 GB
Number of output files	1
Output DST size	500 MB
Number of events	*90M*
Number of jobs	*250*
Input data size	*18 TB*
Output data size	*125 GB*

Usage of SRM

- LHCb CLI tools
  - Stage request
  - File status
  - Advisory delete
- CLI tools built on GFAL library - aim to avoid any SRM version dependencies

SRM (vsn 1.1) Experience with CASTOR

- inability to pin/unpin or mark file for garbage collection - poss. workarounds (redefined SRM "advisory delete" provided)
  - Throttle jobs - manpower intensive (not feasible)
  - New SRM stage request at each file check - use on LCG
  - Technology specific commands - use on LXBATCH for debugging workflow
  - SRM "advisory delete" re-defined
- SRM fails to deal with corrupted/missing files
  - If error returned to SRM all subsequent files are also marked as fail (even if successful!) - needs new CASTOR implementation

SRM (vsn 1.1) Experience with CASTOR

- No control over stage pool - mixing of general user & prod manager
  - Solved - LCG can now check on user and responsibility and assign pool accordingly
- Access rights
  - if one server creates files under one user account, it is not readable by the other servers if the mapping is to another user - problem solved

Step-by-step through stripping use case

1.Production manager reserves the needed pool space at the Tier-1 centres to receive output from concurrent jobs

2.Production manager launches production jobs

3.Production job via SRM checks the VO specific namespace (without the need to know the site specific high level details) if the necessary directory structure exists at the production Tier-1 and the other Tier-1's for the output files. If not will create the necessary directory hierarchy everywhere

4.Check to see if output file already exists at any Tier-1. If so exits with warning message to production manager. If file/directory exists at only 1 site it may be necessary for the production manager to delete/copy the offending file/directory elsewhere.

5.Job issues stage request via SRM for all needed input files

6.As files become available they are pinned by SRM with a validity time compatible with the

7.Jo

8.Or

9.Jo

rea

10.C

Tier-1's.

Need the reserve space management & reservation functionality - necessary to ensure the stripped DSTs have storage at LHCb Tier-1's. Likely to be a slight overestimate &/or usage of "space update"

1.Production manager reserves the needed pool space at the Tier-1 centres to receive output from concurrent jobs

2.Production manager launches production jobs

3.Production job via SRM checks the VO specific namespace if the necessary directory structure exists at the production Tier-1 and the other Tier-1's for the output files. If not will create the necessary directory hierarchy everywhere

4.Check to see if output file already exists at any Tier-1. If so exits with warning message to production manager. If file/directory exists at only 1 site it may be necessary for the production manager to delete/copy the offending file/directory elsewhere.

5.Job issues stage request via SRM for all needed input files

6.As
the

7.Jo

8.Ou

9.Jo
read permission for the entire VO

Need basic directory functionality - necessary to create the hierarchical structure to receive data & check files don't exist etc

10.Once the production is finished the production manager releases reserved space at Tier-1's.

1.Production manager reserves the needed pool space at the Tier-1 centres to receive output from concurrent jobs

2.Pro

3.Pro

struc

not will create the necessary directory hierarchy everywhere

4.Check to see if output file already exists at any Tier-1. If so exits with warning message to production manager. If file/directory exists at only 1 site it may be necessary for the production manager to delete/copy the offending file/directory elsewhere.

5.Job issues stage request via SRM for all needed input files

6.As files become available they are pinned by SRM with a validity time compatible with the expected duration of the job.

7.Job processes files as they become available, once processed the file will be unpinned

8.Once all (available) input files are processed the output file(s) are made permanent

9.Job checks the permission on the file to ensure against accidental deletion but to allow read permission for the entire VO

10.Once the production is finished the production manager releases reserved space at Tier-1's.

Current SRM v1.1 functionality - ability to optimise use of MSS system

1.Production manager reserves the needed pool space at the Tier-1 centres to receive output from concurrent jobs

2.Pro

3.Pro

struc

not w

4.Che

mess

necessary for the production manager to delete/copy the offending file/directory elsewhere.

5.Job issues stage request via SRM for all needed input files

6.As files become available they are pinned by SRM with a validity time compatible with the expected duration of the job.

7.Job processes files as they become available, once processed the file will be unpinned

8.Once all (available) input files are processed the output file(s) are made permanent

9.Job checks the permission on the file to ensure against accidental deletion but to allow read permission for the entire VO

10.Once the production is finished the production manager releases reserved space at Tier-1's.

Many jobs will be running in parallel - important for them not to interfere with each other. Essential to pin file once staged to ensure its availability (& unpin, of course!)

1.Production manager reserves the needed pool space at the Tier-1 centres to receive output from concurrent jobs

2.Pro

3.Pro

struc

not w

4.Che

message to production manager. If file/directory exists at only 1 site it may be necessary for the production manager to delete/copy the offending file/directory elsewhere.

5.Job issues stage request via SRM for all needed input files

6.As files become available they are pinned by SRM with a validity time compatible with the expected duration of the job.

7.Job processes files as they become available, once processed the file will be unpinned

8.Once all (available) input files are processed the output file(s) are made permanent

9.Job checks the permission on the file to ensure against accidental deletion but to allow read permission for the entire VO

10.Once the production is finished the production manager releases reserved space at Tier-1's.

Current SRM v1.1 functionality - ability to store files in MSS system. Necessary before a reserved space is released!

1.Production manager reserves the needed pool space at the Tier-1 centres to receive output from concurrent jobs

2.Pro

3.Pro

struc

not w

4.Che

mess

Permissions functions - essential to ensure file is readable by whole VO and only production manager has write access. Specialised stripping would need to set group/sub-group privileges

necessary for the production manager to delete/copy the offending file/directory elsewhere.

5.Job issues stage request via SRM for all needed input files

6.As files become available they are pinned by SRM with a validity time compatible with the expected duration of the job.

7.Job processes files as they become available, once processed the file will be unpinned

8.Once all (available) input files are processed the output file(s) are made permanent

9.Job checks the permission on the file to ensure against accidental deletion but to allow read permission for the entire VO

10.Once the production is finished the production manager releases reserved space at Tier-1's.

1.Production manager reserves the needed pool space at the Tier-1 centres to receive output from concurrent jobs

2.Pro

3.Pro

struc

not w

4.Che

message to production manager. If file/directory exists at only 1 site it may be necessary for the production manager to delete/copy the offending file/directory elsewhere.

5.Job issues stage request via SRM for all needed input files

6.As files become available they are pinned by SRM with a validity time compatible with the expected duration of the job.

7.Job processes files as they become available, once processed the file will be unpinned

8.Once all (available) input files are processed the output file(s) are made permanent

9.Job checks the permission on the file to ensure against accidental deletion but to allow read permission for the entire VO

10.Once the production is finished the production manager releases reserved space at Tier-1's.

Again use of space management functions - needed for releasing the space used for the output of stripping.

Summary

- All expts see SRM as essential
  - Fundamental building block of a datagrid
- LCG set of functionality - agreed by all expts
- Optimisation of MSS essential
  - Communicate multiple requests
  - Handle priorities
  - Optimise tape access

Storage system manager reqts but recognised by expts

- Needs to be exposed to expt before SC4
  - Timescales tight