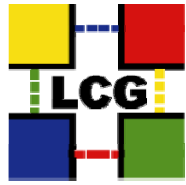


# Storage Resource Manager version 2.2: design, implementation, and testing experience

CHEP 2007 - Victoria, CANADA  
2-9 September 2007

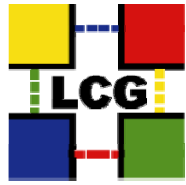
Flavia Donno  
IT/GD, CERN  
*for the SRM v2.2 collaboration*





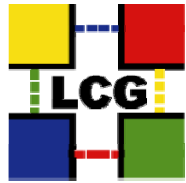
## Outline

- Storage Services in WLCG
- The requirements for a Storage protocol
- The Storage Resource Manager v2.2
- Protocol definition activities
- The study of the SRM v2.2 specification
- Available implementations and test suites
- The S2 testing framework
- Test results
- The GLUE Schema
- The GSSD working group
- Conclusions



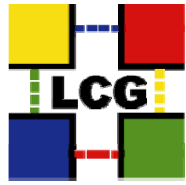
## Storage Services in WLCG

- *Storage Services* are *crucial* components of the *Worldwide LHC Computing Grid* (WLCG) infrastructure spanning more than 200 sites and serving computing and storage resources to the High Energy Physics LHC communities.
- Up to *tens of Petabytes of data* are collected every year by the 4 LHC experiments at CERN.
- To process these large data volumes it is important to *establish a protocol* and a very efficient interface to the various storage solutions adopted by the WLCG sites.
- In this work *we report on the experience* acquired during the definition, implementation, verification and validation of the Storage Resource Manager (SRM) v2.2 protocol



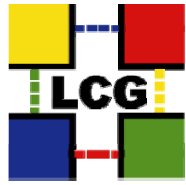
## Requirements definition by dates

- In *June 2005* the *Baseline Service Working Group* published a report:
  - <http://lcg.web.cern.ch/LCG/peb/bs/BSReport-v1.0.pdf>
  - A *Storage Element Service* is mandatory and high priority.
  - The *experiment requirements* for a Grid storage service are defined
  - The full set of recommended feature available by *February 2006*
  - Experiments agree to use only high-level tools as interface to SRM
- The report was *based on the early experience* acquired with SRM v1.1 and v2.1 (never deployed).
- *Mumbai workshop (CHEP2006)*: the experiments had learned more about what was needed and changed their requirements.
- In *May 2006* at FNAL the WLCG SRM Memorandum of Understanding (MoU) was agreed on:
  - <http://cd-docdb.fnal.gov/0015/001583/001/SRMLCG-MoU-day2%5B1%5D.pdf>



## Basic requirements

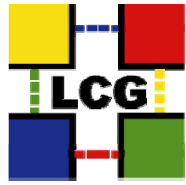
- Support for **Permanent files** (and volatile copies)
- Support for **Permanent Space**
- **Space Reservation** : static only per VO.
  - If dynamic space reservation is available, allow for the possibility of releasing the allocated space
- **Permission Functions** only on directories based on VOMS group/roles
- **Directory Management Functions**
- **Data Transfer and File Removal Functions**
- File access **protocol negotiation**
- VO-specific **relative paths**



## The Storage Resource Manager SRM v2.2

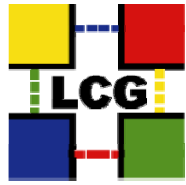
- The *Storage Resource Manager* (SRM) is a **middleware component** whose function is to provide **dynamic space allocation** and **file management** on shared storage components **on the Grid**.
- More precisely, the SRM is a **Grid service** with several different implementations. Its main specification documents are:
  - A. Sim, A. Shoshani (eds.), **The Storage Resource Manager Interface Specication, v. 2.2**, available at <http://sdm.lbl.gov/srm-wg/doc/SRM.v2.2.pdf>.
  - F. Donno et al., **Storage Element Model for SRM 2.2 and GLUE schema description, v3.5** available at: <http://glueschema.forge.cnaf.infn.it/uploads/Spec/V13/SE-Model-3.5.pdf>





# The SRM Interface in a nutshell

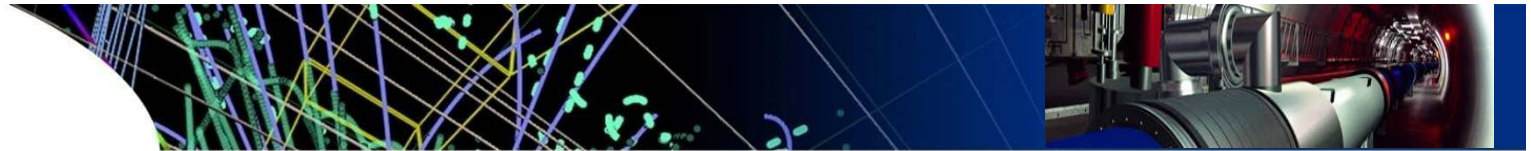
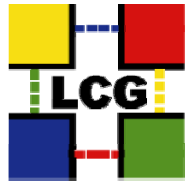
- The SRM Interface Specification lists the *service requests*, along with the *data types* for their arguments.
- Function signatures are given in an implementation-independent language and grouped by functionality:
  - *Space management functions* allow the client to reserve, release, and manage spaces, their types and lifetimes. Support for different qualities of storage space.
    - Reserve/Release Space, ChangeSpaceForFiles, ExtendFileLifeTimeInSpace
  - *Data transfer functions* have the purpose of getting files into SRM spaces either from the client's space or from other remote storage systems on the Grid, and to retrieve them.
    - PrepareToPut/StatusOfPutRequest/PutDone
    - PrepareToGet/StatusOfGetRequest
    - BringOnline
    - Copy
    - ReleaseFiles, AbortRequest/Files, ExtendFileLifeTime
  - *Other function classes* are *Directory, Permission, and Discovery functions*.
    - Ping, Ls, Mkdir, Rm, Rmdir, SetPermission, CheckPermission, etc.



## SRM 2.2 protocol definition activities

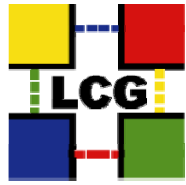
- *Long protocol definition process*: many discussions on the concepts of spaces, file copies management, name space handling.
- *Storage services providers* had to be heavily involved in the discussions.
- From May to December 2006 *weekly meetings* to follow the development.
- *Formal static and dynamic protocol models* defined: identifying unanticipated behaviours and interactions.
  - F. Donno, A. Domenici, *A Model for the Storage Resource Manager*, ISGC 2007, International Symposium on Grid Computing, Taipei, March 26 . 29, 2007





## SRM 2.2 protocol definition activities

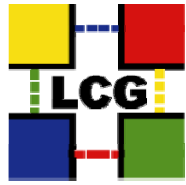
- Need for *testing suites to validate the protocol* against real use cases and to verify the *compliance of the implementations* against the spec.
- The testing campaign motivated the developers to *reconsider many of the initial assumptions and decisions*, leading to solutions that seem to better satisfy the needs of the users.
- The *LBNL testing suite* written in Java was the only one available till September 2006, run manually. Since January 2007 running automatically every day.
  - <http://sdm.lbl.gov/srm-tester/v22-progress.html>
  - <http://sdm.lbl.gov/srm-tester/v22daily.html>
- In September 2006 CERN took over from RAL the development of the *S2 SRM 2.2 testing suite*, enhancing it with a complete test set and with publishing and monitoring tools.
- *Reports to WLCG Management Board* to monitor progress.



## Study of SRM 2.2 specification

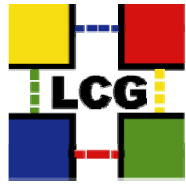
- In September 2006 *very different interpretations* of the spec
- *6 Releases of the SRM v2.2* specification document: July, September, December 2006 and January, April 2007
- Study of the spec (*state/activity diagrams*): many unspecified behaviours.
- A list of about *50 open issues* has been compiled in *September 2006*.
- Last *30 points discussed* and agreed during the *WLCG Workshop* in *January 2007*. Other major points delayed to SRM 3.0.
- *The study of the specifications*, the discussions and testing of the open issues have helped *insure coherence* in the protocol definition and *consistency between SRM implementations*.

<https://twiki.cern.ch/twiki/bin/view/SRMDev/IssuesInTheSpecifications>



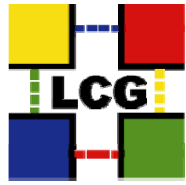
## Storage providers involvement: the available implementations

- SRM v2.2 implementations are available today for the following Storage Services:
  - *CASTOR2* : Hierarchical Storage Server (HSS). Developed by CERN and RAL. SRM v2.2 support in v.2.1.4.
  - *dCache* : HSS developed by DESY and FNAL. SRM 2.2 support in v1.8.
  - *DPM* : disk-only developed by CERN. SRM v2.2 support in v1.6.5 in production.
  - *StoRM* : disk-only developed by INFN and ICTP. SRM v2.2 interface for many filesystems: GPFS, Lustre, XFS and POSIX generic filesystem. SRM v2.2 support in v1.3.15.
  - *BeStMan* : disk-based developed by LBNL. SRM v2.2 support in v2.2.0.0.



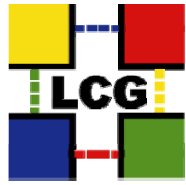
## Need for tests: the black box testing

- Functional testing used for:
  - *Validation*: against the explicit and implicit user requirements
  - *Consistency*: checking for inconsistency, incompleteness, or inefficiency
  - *Verification*: correctness of implementations with respect to the specification
  - *Performance*: ... and reliability
- *Exhaustive test* can demonstrate the correctness of SUT. But it is impracticable.



## Need for tests: limiting the number of test cases

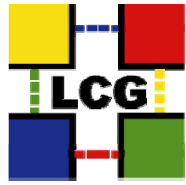
- The *specifications are examined* to find:
  - Input domain of the system
  - Operating conditions that affect the system behaviour
  - Input-output relationship that defines how the system reacts to inputs under given operating conditions.
- Several (black box) testing techniques to *limit the number of test cases*:
  - Equivalence partitioning
  - Boundary-value testing
  - Cause-effect graphing
  - Error guessing



## The S2 testing framework

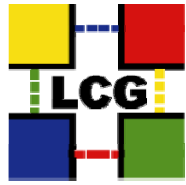
- We needed a *testing framework* with the following characteristics:
  - Support for quick development of single test cases
  - Minimization of human coding errors
  - Plug-in mechanisms for external libraries (such as an SRM client)
  - Engine for parsing the output of a command
  - Support for parallel and sequential execution as well as timers both at instruction and test unit level.
  - “self-describing” logging facility
- Examined *several products available* on the market (TTCN-3 standard for telecommunication protocols)
- We adopted the *S2 tree-based testing language* developed initially at RAL.
  - <http://s-2.sourceforge.net>





## The S2 test families

- The S2 test suite families.
  - Availability: Ping and full put cycle
  - Basic: basic functionality, checking only return codes and passing of basic input parameters
  - Usecases: testing boundary conditions, exceptions, real use cases extracted from the middleware clients and experiment applications.
  - Interoperability: servers acting as clients for put and get requests, cross copy operations
  - Exhaustive: Checking for long strings, strange characters in input arguments, missing mandatory or optional arguments. Output parsed.
  - Stress: Parallel tests for stressing the systems, multiple requests, concurrent colliding requests, space exhaustion, etc.
- S2 tests cron job running *5 times per day* on *21 SRM v2 endpoints*/5 storage flavours with *different configurations*.
- The S2 test suite has been integrated in the *WLCG certification process*
- *S2* has been *used* as well *by the developers* to check constantly their implementation and *distributed to OSG* for Tier-2 validation.
- Test families *continuously expanded* to cover new scenarios, new decisions taken at the protocol level, and check for discovered bugs or race conditions.
- *Integration with WLCG high-level tools* is ongoing.



## Tests executed

- *Results published* on a web page. Latest and history available:
  - <https://twiki.cern.ch/twiki/bin/view/SRMDev>
- Results of failed and successful tests *reported to the developers* to signal issues. A status page is kept up-to-date:
  - <https://twiki.cern.ch/twiki/bin/view/SRMDev/ImplementationsProblems>
- Test results and issues are *discussed on srm-tester and srm-devel lists*
  - <https://hpcrdm.lbl.gov/mailman/listinfo/srmtester>
  - <http://listserv.fnal.gov/archives/srm-devel.html>



# The S2 result web-pages: Basic tests

Summary of S2 SRM v2.2 basic tests - Mozilla Firefox

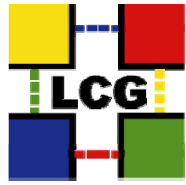
File Edit View History Bookmarks Tools Help

Summary of S2 SRM v2.2 basic test - Thursday 30 August 2007 11:46pm CEST

SRM function	CERN C2	CNAF C2	CERN C2-1	BNL2 dCache	DESY dCache	UKED dCache	FZK dCache	IN2P3 dCache	NDGF dCache	SARA dCache	FNAL dCache	UCSD dCache	CERN DPM	UKED DPM	UKGL DPM	LAL DPM	LBNL BeStMan	CNAF StoRM	CNAF StoRM2	UKBR StoRM	IFIC StoRM
<b>WLCG MoU SRM v2.2 methods</b>																					
<a href="#">Ping</a>	Out Log	Out Log	Out Log	Out Log	Out Log	Out Log	Out Log	Out Log	Out Log	Out Log	Out Log	Out Log	Out Log	Out Log	Out Log	Out Log	Out Log	Out Log	Out Log	Out Log	Out Log
<a href="#">PTP</a>	Out Log	Out Log	Out Log	Out Log	Out Log	Out Log	Out Log	Out Log	Out Log	Out Log	Out Log	Out Log	Out Log	Out Log	Out Log	Out Log	Out Log	Out Log	Out Log	Out Log	Out Log
<a href="#">StoRPut</a>	Out Log	Out Log	Out Log	Out Log	Out Log	Out Log	Out Log	Out Log	Out Log	Out Log	Out Log	Out Log	Out Log	Out Log	Out Log	Out Log	Out Log	Out Log	Out Log	Out Log	Out Log
<a href="#">PutDone</a>	Out Log	Out Log	Out Log	Out Log	Out Log	Out Log	Out Log	Out Log	Out Log	Out Log	Out Log	Out Log	Out Log	Out Log	Out Log	Out Log	Out Log	Out Log	Out Log	Out Log	Out Log
<a href="#">PtG</a>	Out Log	Out Log	Out Log	Out Log	Out Log	Out Log	Out Log	Out Log	Out Log	Out Log	Out Log	Out Log	Out Log	Out Log	Out Log	Out Log	Out Log	Out Log	Out Log	Out Log	Out Log
<a href="#">StoRGet</a>	Out Log	Out Log	Out Log	Out Log	Out Log	Out Log	Out Log	Out Log	Out Log	Out Log	Out Log	Out Log	Out Log	Out Log	Out Log	Out Log	Out Log	Out Log	Out Log	Out Log	Out Log
<a href="#">BoL</a>	Out Log	Out Log	Out Log	Out Log	Out Log	Out Log	Out Log	Out Log	Out Log	Out Log	Out Log	Out Log	Out Log	Out Log	Out Log	Out Log	Out Log	Out Log	Out Log	Out Log	Out Log
<a href="#">StoRBoL</a>	Out Log	Out Log	Out Log	Out Log	Out Log	Out Log	Out Log	Out Log	Out Log	Out Log	Out Log	Out Log	Out Log	Out Log	Out Log	Out Log	Out Log	Out Log	Out Log	Out Log	Out Log
<a href="#">AbortR</a>	Out Log	Out Log	Out Log	Out Log	Out Log	Out Log	Out Log	Out Log	Out Log	Out Log	Out Log	Out Log	Out Log	Out Log	Out Log	Out Log	Out Log	Out Log	Out Log	Out Log	Out Log
<a href="#">AbortF</a>	Out Log	Out Log	Out Log	Out Log	Out Log	Out Log	Out Log	Out Log	Out Log	Out Log	Out Log	Out Log	Out Log	Out Log	Out Log	Out Log	Out Log	Out Log	Out Log	Out Log	Out Log
<a href="#">RetFiles</a>	Out Log	Out Log	Out Log	Out Log	Out Log	Out Log	Out Log	Out Log	Out Log	Out Log	Out Log	Out Log	Out Log	Out Log	Out Log	Out Log	Out Log	Out Log	Out Log	Out Log	Out Log
<a href="#">GetReqSum</a>	Out Log	Out Log	Out Log	Out Log	Out Log	Out Log	Out Log	Out Log	Out Log	Out Log	Out Log	Out Log	Out Log	Out Log	Out Log	Out Log	Out Log	Out Log	Out Log	Out Log	Out Log
<a href="#">GetReqToks</a>	Out Log	Out Log	Out Log	Out Log	Out Log	Out Log	Out Log	Out Log	Out Log	Out Log	Out Log	Out Log	Out Log	Out Log	Out Log	Out Log	Out Log	Out Log	Out Log	Out Log	Out Log
<a href="#">GetTrProts</a>	Out Log	Out Log	Out Log	Out Log	Out Log	Out Log	Out Log	Out Log	Out Log	Out Log	Out Log	Out Log	Out Log	Out Log	Out Log	Out Log	Out Log	Out Log	Out Log	Out Log	Out Log
<a href="#">Le</a>	Out Log	Out Log	Out Log	Out Log	Out Log	Out Log	Out Log	Out Log	Out Log	Out Log	Out Log	Out Log	Out Log	Out Log	Out Log	Out Log	Out Log	Out Log	Out Log	Out Log	Out Log







# The S2 result web-pages: Availability, interoperability and stress

Summary of S2 SRM v2.2 availability tests - Mozilla Firefox

File Edit View History Bookmarks Tools Help

Summary of S2 SRM v2.2 availability test - Friday 31 August 2007 12:10am CEST

CERN C2	CERN C2-1	BNL dCache	DESY dCache	UCL dCache	FNAL dCache	IN2P3 dCache	NDGF dCache	RAPA dCache	FNAL dCache	UCSD DPM	CERN DPM	UCL DPM	UCL DPM	LAL BeStMan	LBNL dCache	CNAF dCache	CERN dCache	DESY dCache	FNAL dCache	
UP	UP	DO WH	UP	DO WH	UP	UP	UP	UP	UP	UP	UP	UP	UP	UP	UP	UP	UP	DO WH	DO WH	DO WH

Summary of S2 SRM v2.2 cross tests - Mozilla Firefox

File Edit View History Bookmarks Tools Help

Summary of S2 SRM v2.2 cross test - Wednesday 15 August 2007 01:30am CEST

In these tests the smCopy function is exercised. This function should be implemented by all available Storage System by the end of the 3Q of 2007. dCache is required to implement this function as of now. Therefore, it is OK to have red columns for all SRM endpoints except for dCache. However, it is not OK to have red rows since this means that a file cannot be copied between SRMs with simple get and put operations.

SRM function	CERN C2	DESY dCache	FNAL dCache	CERN DPM	LBNL BeStMan	CNAF dCache
<b>Copy Tests in PUSH mode</b>						
CopyToCERNCASTOR	Out Log	Out Log	Out	Out Log	Out Log	Out Log
CopyToFNALDCACHE	Out Log	Out Log	Out	Out Log	Out Log	Out Log
CopyToDESYDCACHE	Out Log	Out Log	Out	Out Log	Out Log	Out Log
CopyToCERNNDPM	Out Log	Out Log	Out	Out Log	Out Log	Out Log
CopyToLBNLDRM	Out Log	Out Log	Out	Out Log	Out Log	Out Log
CopyToSTORM	Out Log	Out Log	Out	Out Log	Out Log	Out Log
<b>Copy Tests in PULL mode</b>						
CopyFromCERNCASTOR	Out Log	Out Log	Out Log	Out Log	Out Log	Out Log
CopyFromFNALDCACHE	Out Log	Out	Out	Out	Out Log	Out
CopyFromDESYDCACHE	Out Log	Out Log	Out Log	Out Log	Out Log	Out Log
CopyFromCERNNDPM	Out Log	Out Log	Out Log	Out Log	Out Log	Out Log
CopyFromLBNLDRM	Out Log	Out Log	Out Log	Out Log	Out Log	Out Log
CopyFromSTORM	Out Log	Out Log	Out Log	Out Log	Out Log	Out Log

Summary of S2 SRM v2.2 stress tests - Mozilla Firefox

File Edit View History Bookmarks Tools Help

Summary of S2 SRM v2.2 stress test - Monday 25 June 2007 03:02pm CEST

SRM test	CERN DPM
GetParallel	133
	438
	789

Configuration parameters

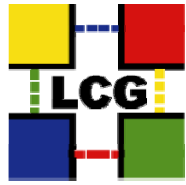
```
# Number of Threads
Export M_THREADS 50
# Number of (bulk) operations
Export N_OPS 50
# Polling frequency
Export SLEEP_SOR 2 # sec (Status of Request)
# Looping
Export LOOP 200
```

Summary of S2 SRM v2.2 stress test - Monday 25 June 2007 03:10pm CEST

SRM test	CERN DPM
GetParallel	123
	456
	789

Configuration parameters

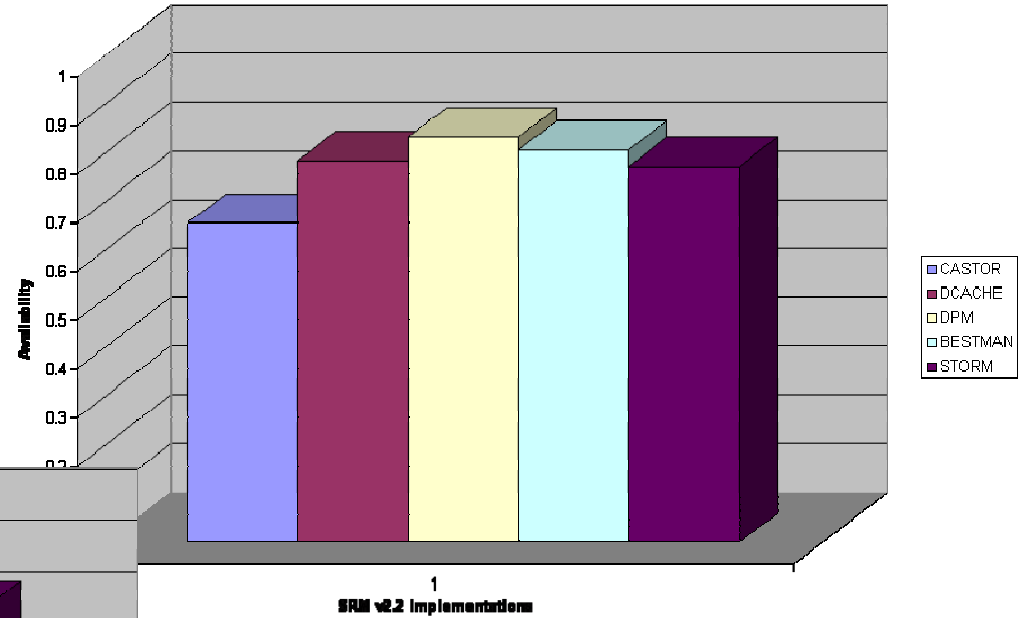
```
# Number of Threads
Export M_THREADS 70
# Number of (bulk) operations
Export N_OPS 70
# Polling frequency
Export SLEEP_SOR 2
# Looping
Export LOOP 200
```



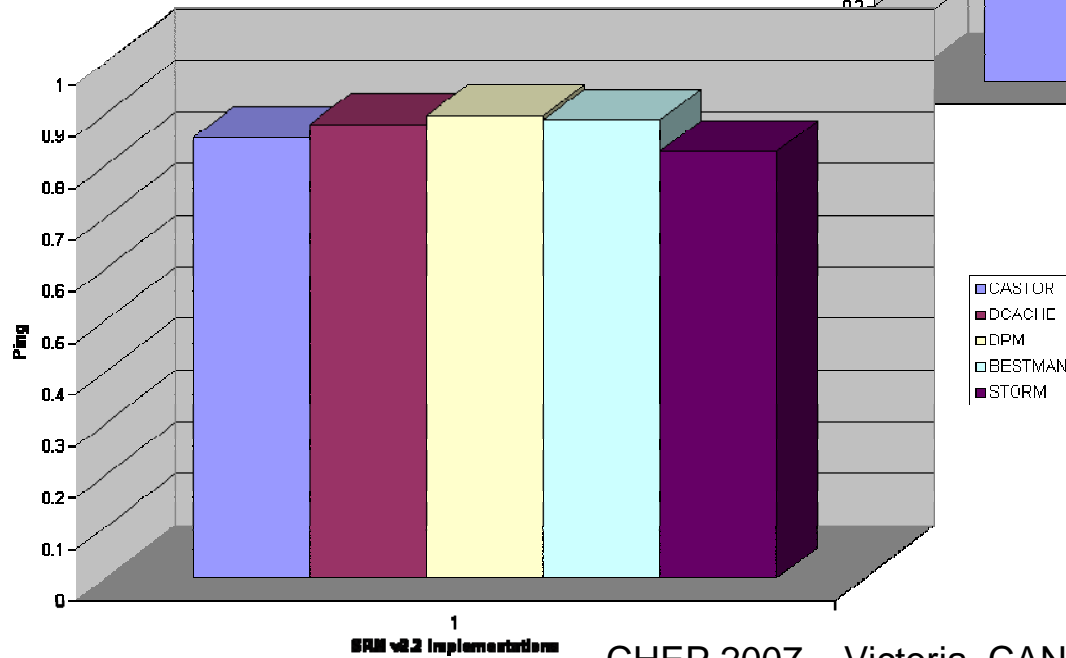
# Test results



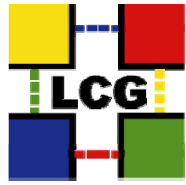
Availability: Period 16/12/2006 - 30/09/2007



Ping: Period 16/12/2007 - 30/09/2007

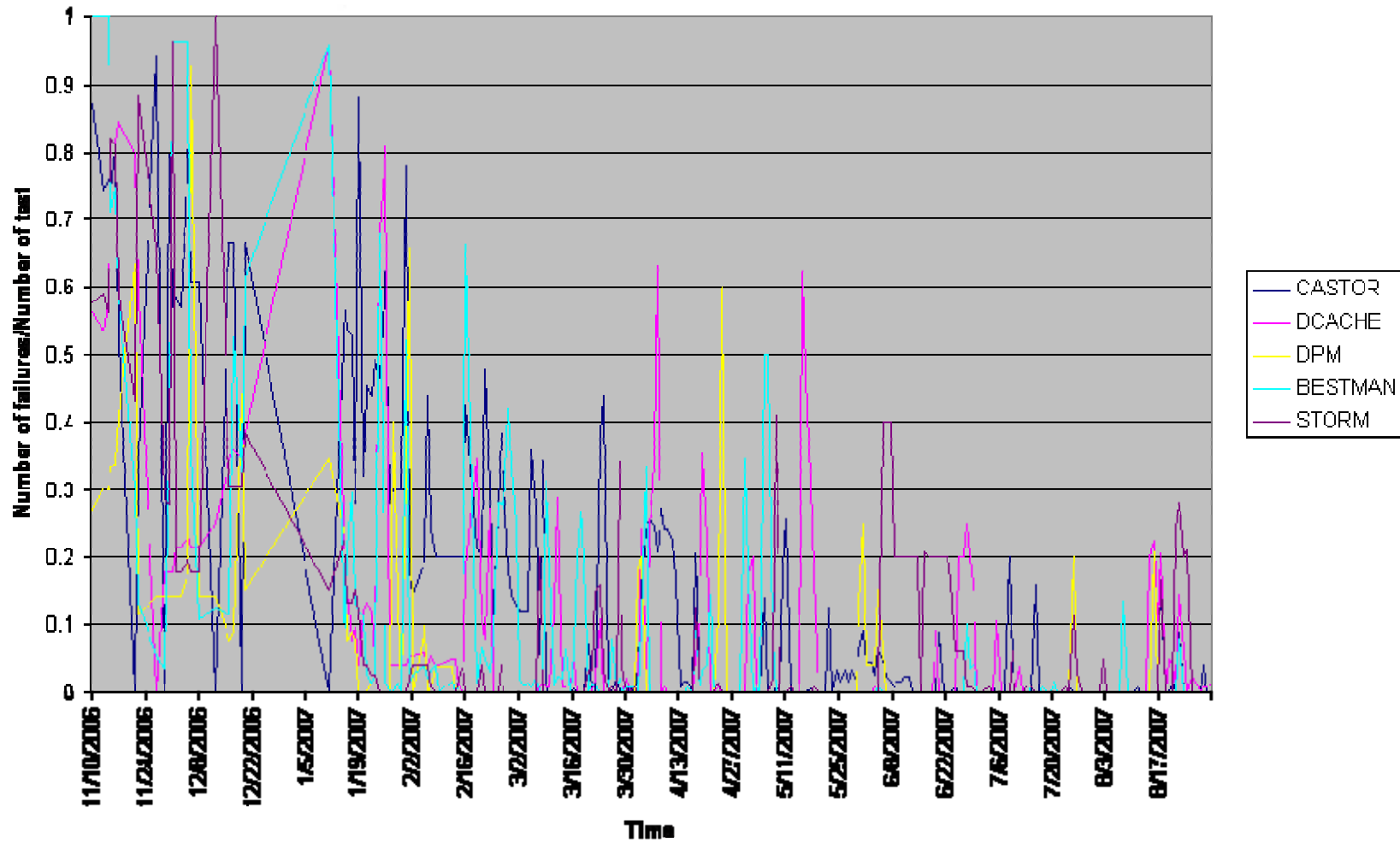


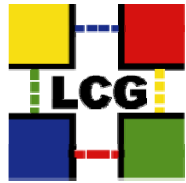




Basic: Period 10/11/2005 - 30/08/2007

# Test results

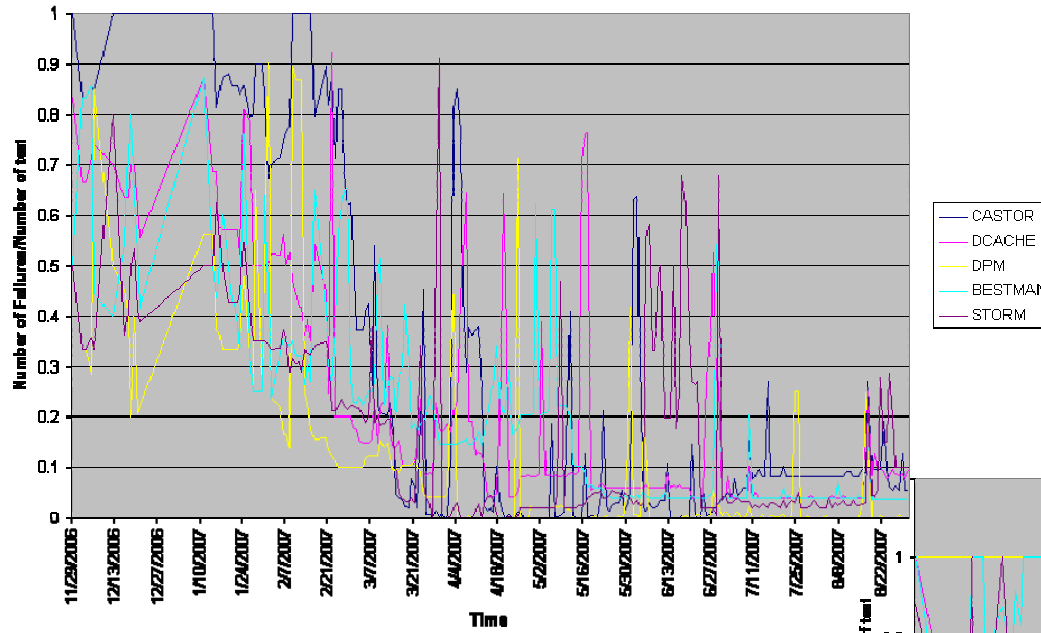




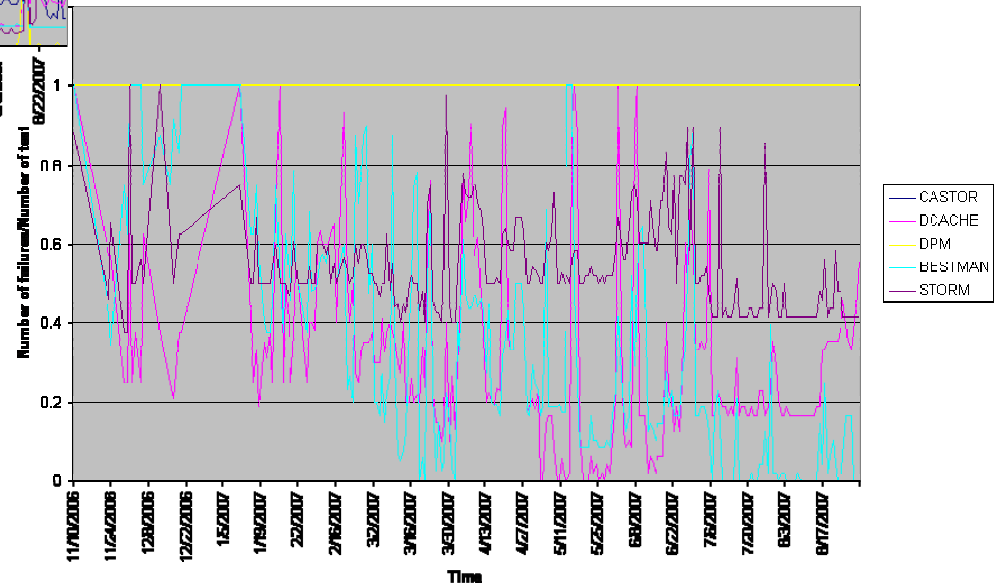
Use Case: Period 29/11/2006 - 30/06/2007

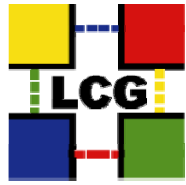


# Test results



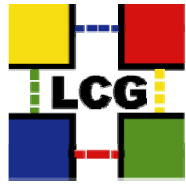
Interoperability: Period 10/11/2006 - 30/06/2007





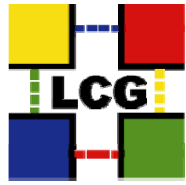
## Other testing efforts: the SRM clients

- Other testing groups to:
  - Check the status of the high-level SRM v2 clients such as GFAL, lcg-utils, FTS
  - Transparent access to storage independent of the protocol in use (SRM v1 vs SRM v2).
  - Experiment data access patterns.
  - Correct handling of VO specific proxies with VOMS groups/roles.
  - Compliance to user requirements and functionalities needed
  - Availability of required bindings (C++, Python, Perl, etc.) and support for mandatory platforms (SL4 32-bit mode, 64-bit mode later).
- Outcomes:
  - User documentation.
  - User education and training.
  - Preparation to the integration of experiment frameworks with the SRM v2 environment.



## GLUE Schema

- The WLCG information system publishes details about grid services in a format that is specified by the GLUE schema.
- GLUE 1.3 includes information needed for SRM v2.2, such as the space information.
  - <http://glueschema.forge.cnaf.infn.it/Spec/V13>
- The modelling and testing exercises have helped in the specifications of GLUE 1.3 for Storage Services.
- The current experience has also been taken as a base for the design of the new GLUE 2.0 model for Storage.



## Grid Storage System Deployment (GSSD)

- <https://twiki.cern.ch/twiki/bin/view/LCG/GSSD>
- Mailing list: [storage-class-wg@cern.ch](mailto:storage-class-wg@cern.ch)
- Mandate:
  - Coordinating the deployment of SRM v2.2 in WLCG with sites, experiments, and developers.
  - Establishing a migration plan from SRM v1 to SRM v2 to ensure smooth operations during the transition.
  - Coordinating the provision of the necessary information by the Storage Providers in order to publish and monitor the status of storage resources, guaranteeing the availability of the requested resources and their correct usage.
  - Ensure transparency of data access and the functionalities required by the experiments.



## Conclusions

- The SRM specification definition and implementation process has evolved in a *world-wide collaboration effort* with developers, independent testers, experiments and site administrators.
- Much *clearer* description of SRM *specifications*. A few issues are left out for SRM v3.
- Well *established* and agreed *methodology* to validate a protocol and verify the status of the implementations.
- *Clear plan* to make sure that experiment requirements are covered and the proposed implementations respond to the expectation also in terms of performance.
- The modelling and testing exercises have helped in the specifications of Storage Services in the *Grid Information System*.
- The GSSD working group is in charge of the *deployment plan* for SRM v2.2 in production on the WLCG infrastructure, to be ready for LHC operations.