

StoRM

INFN Solution for Storage Resource Manager in Grid

Alberto Forti, Luca Magnoni, Riccardo Zappi
INFN-CNAF

CCR and INFN-GRID Workshop, 2009

11-15 May 2009, Palau

- SRM in one slide
- StoRM in a nutshell
- StoRM v1.4.x
- Weather forecast
- Conclusions



The SRM service

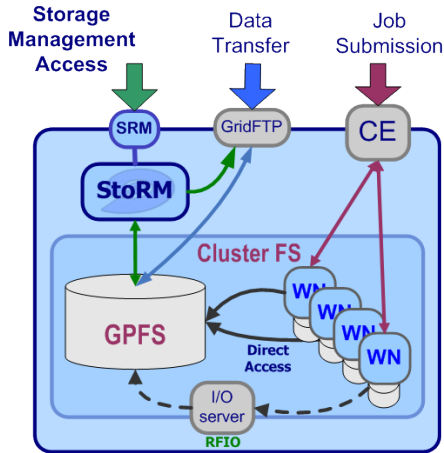
From bibliography:

- **Storage Resource Managers** are middleware services whose function is to provide dynamic space allocation and file management of shared storage components.
- Files are no longer permanent entities on the storage, but **dynamic** ones that can appear or disappear according to the user's specification.
- SRMs do not perform file transfers, but can invoke middleware components that perform this job (such as GridFTP)

StoRM is

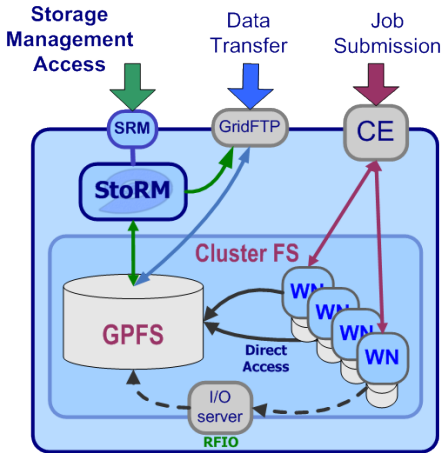
- **A SRM service for different disk(†) based storage systems.**
Sites can change storage system without care about the SRM layer.
- **Simple, configurable and highly scalable.**
Easy enough to be the best substitute for a classicSE and scalable enough to satisfy a Tier1-scaled centre.
- **Efficient.**
High performance on SRM requests execution.
- **Secure.**
Layered security mechanism, VOMS based and highly configurable.

StoRM in a site



- It is designed to take advantage from **high performing cluster file system**, as GPFS and Lustre, but it supports also every standard POSIX File Systems
- It allows **direct access** to the storage resource, as well as other standard grid protocol as *gsiftp* and *rfio*

StoRM in a site

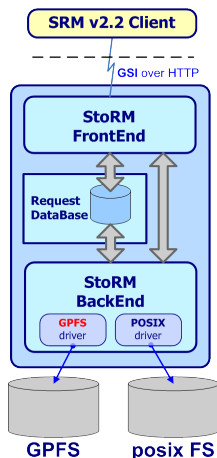


- Authentication and authorization on files and spaces, VOMS-aware
- Permission enforcing are based on setting **physical ACLs** on files and directories

StoRM architecture

StoRM has a multilayer architecture.

- The **Frontend (FE)** component exposes the web service interface
- The **Database** is used to store SRM request data and the internal StoRM metadata
- The **Backend (BE)** is the main component and executes all SRM requests



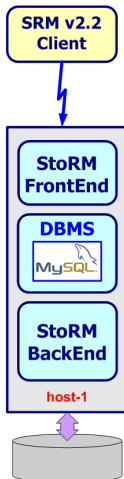
StoRM layouts

StoRM can be deployed in different layout, depending on site requirements and load expected:

- StoRM architecture supports components replication and distribution
- Frontend is cpu intensive **due to GSI auth**. Scale it.
- GridFTP pool managed by DNS or with the new internal load balancer

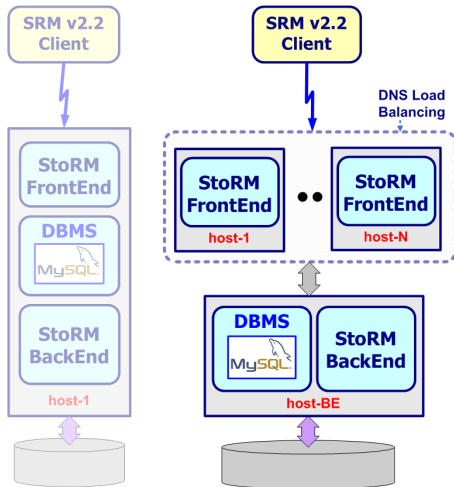
Deployment on different host

- All-in-one installation suitable for small site
- Scaling the number of FEs on the load expected
- Distribute MySQL in case of high performance required



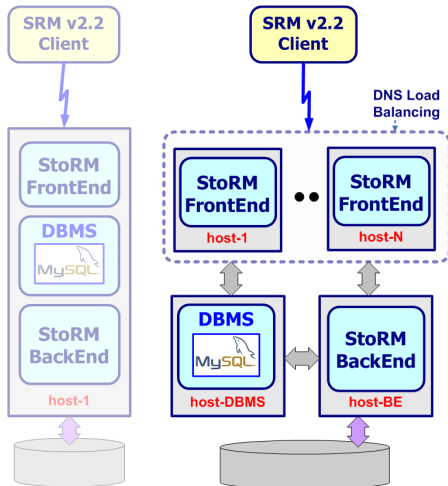
Deployment on different host

- All-in-one installation suitable for small site
- Scaling the number of FEs on the load expected
- Distribute MySQL in case of high performance required



Deployment on different host

- All-in-one installation suitable for small site
- Scaling the number of FEs on the load expected
- Distribute MySQL in case of high performance required



StoRM 1.4.0

New release includes:

- Authorization on Storage Area by user (DN and FQANs regex) and operation types (complex work, test on real use case needed)
- Internal GridFTP pool balancer
- New Frontend thread pool manager
- Improved requests garbage collector
- Checksum support (Adler32)
- New installation directory structure and configuration files
- Many bug fixes
- Changelog [here](#)

StoRM 1.4.x: Space Protection (**Addendum**)

Now, it is possible define ACL on Storage Area

- Shortcuts are enabled, like:
 - EVERYONE: everyone with a valid proxy
- ACL can use Regular Expression to define a set of Subject
 - DN: to define authz policy on Country or Organization for example.
 - FQAN: guidelines from "FQAN Wildcard" document (C.Witzig, EGEE)

StoRM 1.4.x: Internal Pool Balancer on GridFTP

- It is possible define balancing policies on GridFTP
- Some policies:
 - ROUND ROBIN: we don't measure the workload of pool member, we rotate on it.
 - RANDOM: the load is distributed among the pool member in a random fashion.
 - WEIGHT: Each pool member has a weight, the load is distributed according to it.
 - FTP PROCESS MONITOR: The load is distributed depending on the FTP processes running each GridFTP server. The one with less processes is selected. The monitor of the process is based on Farm Management and Control (FMC) system.



StoRM 1.4.x e 1.5.x: the near future

We have some new features planned (ordered by priority)

- 1 Service effectiveness, stability and performance.
 - Security enhancement (Proxy caching)
- 2 Tape support with GPFS and TSM (Italian Tier-1)
- 3 Integration with a Notification mechanism
 - Starting a collaboration with LB team (A.Krenek)
- 3 Service monitoring (like WMS-Monitor)
- 3 Service administration interface



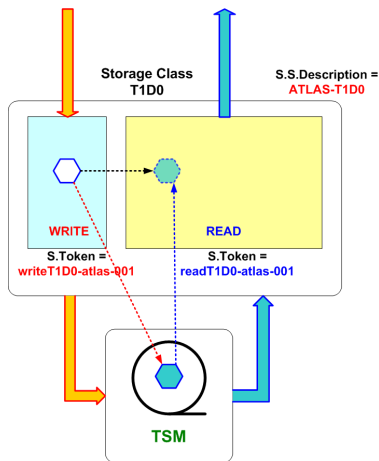
StoRM 1.4.x: Service Effectiveness, Performance

- Support for proxy caching and finger-print (or other tricks to bypass GSI bottleneck)
- Minor updates of the latest release of StoRM
- Bugfixing and minor enhancements (depends on users feedback)

StoRM 1.5.x: Tape support and Space Selection (Addendum)

Support for a hierarchical solution with GPFS and TSM.

- Logic separation between "Staging and Recall buffers"
- User could (eventually) specifies the target Storage Area (token) for Recall operations
- Storage Area will be protected in terms of Authorization and Quota
- We suppose this fullfils your requirements. Right?





StoRM R&D: Notification mechanism

Study, design and develop a "Proof of Concept" to avoid a "*storm*" of Status requests.

- In SRM there are asynchronous requests.
- to retrieve the Status of that requests clients have to query the SRM service.
- The idea is to use a notification mechanism.
- Starting a collaboration with A.Krenek (LB component)



StoRM Usability

- Documentation
- StoRM monitoring service
 - Inspiration (and hopely expertise ;)) from WMSMonitoring tool
- StoRM administration facilities
 - Command line tools to manage StoRM configuration (namespace.xml)

Conclusion

StoRM

- leverages on cluster and parallel file system.
- supports **direct access** on data.
- is **highly configurable**, to satisfy the different site requirements.
- is **flexible**.
 - Scale the Front End if needed.
 - Use any FS you prefer.
- is **lively**.
 - We are working to support Hierarchical Storage System.