



# Busy Storage Services

**Flavia Donno**  
**CERN/IT-GS**

*WLCG Management Board,  
CERN 10 March 2009*

# The problem

---

- **Storage services might become very slow or unresponsive when “busy”, i. e. running at the limit of some resources (high load).**
  - dCache PNFS bottlenecks might make the full dCache service very slow or unresponsive. The situation might degrade so much that the service can become unusable and require human intervention.
- **Current Data Management clients do not handle correctly the situation where the storage service is “busy”.**
  - Some clients (FTS) may [abort and] retry immediately a failed request making the busy status of the storage server even more severe.

**There was no agreed way to communicate the busy status of a server to a client.**

# Goals

---

- We organized 2 phone conferences with storage service and data management clients developers, data management experts and some managers (OSG and EGEE):
  - CASTOR, dCache, DPM, StoRM, BeStMan
  - GFAL/lcg-utils, FTS, dCache srm\* clients, StoRM+BeStMan clients
  - Experiments DM clients
- The goals:
  - To agree on a way to communicate to clients the “busy” status of a server
  - To advise clients on how to react in the presence of a “busy” server



# Conclusions

---

<https://twiki.cern.ch/twiki/bin/view/LCG/CCRC08SSWGStorageBusyMeeting090226>

- The SRM server **MUST** return **SRM\_INTERNAL\_ERROR** when experiencing transient problems (caused by high load on some components).
  - SRM\_FAILURE can always be returned to fail a request.
  
- If the request will be processed the SRM server **SHOULD** return an *estimatedWaitTime* for each file in the request to tell the client when the next polling **SHOULD** happen in order to have a new update on the status of each file.
  
- If the client application receives an **SRM\_INTERNAL\_ERROR** from the SRM server, it **MAY** repeat the request. If it does, it **SHOULD** use a randomized exponential retry time.

■ The algorithm used by Ethernet when congestion is detected was proposed as a good example of randomized exponential retry algorithm.



# Backward-compatibility

- **Current official WLCG Data Management clients can already catch the SRM\_INTERNAL\_ERROR return code**
  - However, they do not act in the optimal way since they might either abort the request or retry immediately
  
- **Current storage services already send the SRM\_INTERNAL\_ERROR code but only in very rare cases (failure of some internal component – i.e. Database)**
  - The return code is sent to the clients when it is already too late
  
- **Therefore, the solution suggested is backward-compatible, although it is advisable but not required for the “busy storage services”-aware **clients to be deployed before** “busy storage services”-enabled **servers** are deployed**
  - Clients might otherwise fail more often than needed



# Status of WLCG DM clients

---

- **The official WLCG Data Management clients that need code changes are:**
  - **FTS** 2.2 will not have any of the agreed changes, and the priority is now on the checksum checks. The coding for the issue at discussion will begin in April, and at least three months are expected for having it in production.
  - For **GFAL/lcg-utils** the implementation of the suggested changes will not start before mid March.
  - We do not have yet a precise schedule for the **dCache srm\*** clients.
  - **StoRM clients** perform only atomic operations at the moment (no retries). Therefore they do not need any changes. Retries will be implemented later on.
  - **BeStMan clients** will be implementing the required changes in the next two weeks.



# Experiment specific DM clients

---

- **We cannot control what experiments do**

- **Reference implementations** are being provided for all known use-cases to provide specific guidelines and examples.
- Follow up with the experiments DM developers
- Make GGUS supporters aware of this work

- **A first reference implementation for the pre-stage use case can be found here:**

- [http://grid-deployment.web.cern.ch/grid-deployment/flavia/prestage\\_release\\_example.py](http://grid-deployment.web.cern.ch/grid-deployment/flavia/prestage_release_example.py)
- Reference implementations for all known use-cases can be ready in about 2 months.



# Status of Storage Servers

---

- The implementation of needed changes can start now for all storage services
- Storage developers will make available a development test endpoint implementing the agreed behavior
- A pre-release of GFAL basic building block functions need to be available in order to check the correct implementation of the storage server against them
- The reference implementations made available for the experiments will be used to test the pre-release of storage service (development test endpoint)
- The S2 stress test suite will be used to make the servers “busy”, causing high load





# Deployment plan

---

- It seems feasible to have both WLCG official Data Management clients and storage services tested and ready for deployment by September 2009
- This date is very close to LHC startup. How should we proceed?



# More information

---

- More information can be found on the **Storage Solution Working Group twiki page**:
  - [https://twiki.cern.ch/twiki/bin/view/LCG/WLCGCommonComputingReadinessChallenges#Storage\\_Solution\\_Working\\_Group\\_S](https://twiki.cern.ch/twiki/bin/view/LCG/WLCGCommonComputingReadinessChallenges#Storage_Solution_Working_Group_S)
  - Please, check the section “Busy Storage Services”

