

## Pending issues for ATLAS

- One file lost at RAL from the CASTOR tape:

during the tests made by Graeme at RAL one file never arrived to be staged:

```
$ lcg-ls -l srm://srm-atlas.gridpp.rl.ac.uk/castor/ads.rl.ac.uk/prod/atlas/raw/atlasdatatape/fdr0
-rwxrwxrw- 1      2      2 1933192856                               NEARLINE
```

Graeme asked Tim Folkes of RAL to investigate.

Tim: DLF is showing errors for recalls for that file. It's getting "Incorrect or missing trailer label on tape". The data part of the file is fine, but the end of tape structure part of the data is missing, so even though the data has been read off the tape, because the trailer is incorrect, it will not mark the disk copy as STAGED, and puts it as INVALID. I'll have to get the data off by hand, and copy it back in as a new file, so the system puts it out to tape again. I have asked the developers to let the code ignore this error, but not sure if they will do anything.

Graeme ask to Shaun: Shaun, is this something the SRM can know about? Can we get a message back so that DDM knows to give up on this file and send the client an error?

Waiting for response (13 Feb 2009)...

- BringOnline lifetime request

ATLAS (namely Graeme) would be very interesting to implement the new feature of the GFAL API to pass the request lifetime as an argument of the BringOnline request. They would like to implement it in their DDM agent and then run again the test, maybe with the preprod version of GFAL They think this would be very useful and should simplify the transition to status polling via the BOL request.

Andrea also investigated as to how long one could query a BOL request and it looks like it's probably a week in CASTOR (this means that formally it doesn't honor the BOL lifetime!).

The understanding of Graeme was that dCache SRM does honor the BOL lifetime and that after this has expired you cannot query its status. Should be clarified.

Andrea confirms: CASTOR does not bother to abort the request after the TotalRequestTime expired. About dCache: he understands that dCache does take into account desiredTotalRequestTime and may abort a request after the TotalRequestTime has expired, but this does not mean that the status of the request cannot be queried after, otherwise you would lose all the information (the request may well have serviced many files). The time after a request disappears is not clearly defined, but it should be at least of the order of days.

TO BE VERIFIED

- OVERWRITE option:

**Simone:** I would like to know if the OVERWRITE option in SRM is tested in the S2 tests (Jean Philippe thinks so). And, in case, I would like to know if it is working for every storage implementation. It would be very useful to have this working for the ATLAS dq2-\* clients which are used especially to upload files on grid storage elements. The other question is for Remi, to understand if lcg-cr and lcg-cp force the OVERWRITE flag or not (or if there is optionally the way to do so), but if you know the answer please tell me.

The OVERWRITE option is interesting for ATLAS for this reason: you try lcg-cp to UPLOAD a file in a SE. This for some reason fails but leaves a zero length file on the storage. At this point, you try again, but the second copy operation fails with a File Exist error. So I was wondering if it would be possible for clients like lcg-cp to force the overwrite.

**Flavia:** it is tested by S2 and if I remember correctly it works for all implementations. More in detail: CASTOR, DPM, BestMan and StoRM return SRM\_FILE\_BUSY as foreseen. dCache behaves differently depending on the version of dCache installed. SARA at the moment returns SRM\_NOT\_SUPPORTED which means that they do not support the OVERWRITE option. For all other dCache sites at the moment I get "operation in progress" and the request stays in queue for long time. Therefore, I do not know what happens. I can try the test by hand and contact the dCache developers.

We had a meeting on this with SRM developers and SRM client application developers. We agreed that in case the user specifies "overwrite" to clients such as FTS and/or lcg-utils and the clients get back from a server a "file exist", the clients (such as FTS and/or lcg\*) should remove the file and try again. We decided not to use the overwrite option in SRM. It was the safest way to go. Please, talk to Paolo, Remi and Akos, since they have open a savannah bug about this to track what they needed to do.

**Andrea:** According to the WLCG addendum, overwriteOption should not be specified: "Files are immutable. WLCG clients shall not specify overwriteOption. If SRM returns failure, the state of the system shall be as if the file transfer did not take place."

**Flavia** Yes, Andrea. But this was long ago and the reality has changed. As I said, we had a meeting where we somehow "overwrote" what we decided in the addendum: The user **CAN** specify the overwrite option to a WLCG client. The client behaves as I said. Please, check for the savannah bug

-- ElisaLanciotti - 13 Feb 2009

---

This topic: LCG > PreStagingTestsPendingIssuesATLAS

Topic revision: r1 - 2009-06-05 - AndreaSciaba



Copyright &© 2008-2020 by the contributing authors. All material on this collaboration platform is the property of the contributing authors.

Ideas, requests, problems regarding TWiki? Send feedback