

Table of Contents

| | |
|---|----------|
| LFC Sessions and Transactions..... | 1 |
| Definitions..... | 1 |
| How to use sessions/transactions..... | 1 |

LFC Sessions and Transactions

Definitions

For more details, refer directly to the `lfc_starttrans` and `lfc_startsess` man pages.

- **Sessions :**

The connection to the Name Server is kept open until `lfc_endsess` is explicitly called. Requests issued between `lfc_startsess` and `lfc_endsess` are independent from each other and are individually committed or rolled back.

- **Transactions :**

The connection to the Name Server is kept open until one of the DB update operation fails or `lfc_aborttrans` or `lfc_endtrans` is explicitly called. If all the DB update operations between `lfc_starttrans` and `lfc_endtrans` are successful, a COMMIT is done. If any DB update operation fails or if an explicit `lfc_aborttrans` is done, a ROLLBACK is done.

How to use sessions/transactions

THIS PART IS A CONTRIBUTION FROM ANTONIO DELGADO PERIS.

- Transactions should only be used for a short set of operations that really need to go together (e.g. create an entry, sets its filesize and add a replica), but not for long activities (and clearly not for deletions or listings). For those things, sessions should be used instead.
- For transactions and sessions, there is a timeout of 60 seconds after the last operation was performed. If no operation is performed in that time, the session/transaction will be closed and an "Internal Error" will occur in next operation.
- Currently the time out for the first operation can be of 5 seconds instead of 60. This will be fixed in LFC/DPM 1.4.1 (all time outs will be of 60 seconds)
- Some calls, like directories opening, listing, listing of replicas, etc. are executed on a different thread (from the transaction/session operations) so they don't count as operations as what regards the 60 seconds timeout.

TO DO : Please specify which operations count and which don't.

- Likewise, some operations use the view of the transaction and some use the normal view. That means that they don't reflect changes that occurred within a non-yet committed transaction. E.g. a directory listing can report about entries that were already deleted.

TO DO : list precisely which methods

- Also, failure in some operations cause a transaction to be rolled back while it doesn't in others. This should be specified for every call. E.g. `lfc_statg` can fail without the transaction is rolled back, but `lfc_access` will cause the rollback to be committed (IMHO, this shouldn't happen, since this call does not update the DB).
- If a directory is opened within a transaction, and modified, a future operation like closing or reading the directory can cause a failure "Internal error", because the DB tries to update the directory while

the previous update (caused by the modification) was not committed yet (because the transaction was not closed yet). This leads us to the point that transactions shouldn't be used for this kind of operations and in particular:

- The transaction should be ended before closing the directory.
- The transaction shouldn't include a long listing of a directory.

-- SophieLemaitre - 22 Nov 2005

This topic: LCG > LfcTransactionsAndSessions

Topic revision: r1 - 2005-11-22 - unknown



Copyright &© 2008-2021 by the contributing authors. All material on this collaboration platform is the property of the contributing authors.

or Ideas, requests, problems regarding TWiki? use [Discourse](#) or [Send feedback](#)