



lustre®

Simplified Interop Recovery

Huang Hua

Lustre Group

Sun Microsystems, Inc.

Motivations

Interoperability between releases

upgrade/downgrade as a normal failover

Wire protocol changes from 1.8 to 2.0

replay/resend request, reply

Request need reformat

Reduce/eliminate the request reformat, simplify the implementation and testing

Not only for interop, but also for normal failover

High Level Design

Reduce the replay/resend if possible

- Controlled failover

Notify the clients when a server is going to shut down

- Transaction commit

- Cache flush

- Blocking new requests to that server

DLM lock: LCK_EX on server, LCK_CR on clients.

Server notifies the clients by AST

Current Status

Not yet finished

Code Inspection is undergoing

Preliminary testing shows that it works

Targeted for 1.8.1 and 2.0

Details and Focus for Inspection (1)

New connection flag : OBD_CONNECT_UPDATE_LOCK

Maintaining interop

mdc-mds DLM lock is a LDLM_IBITS lock: mds only supports IBITS lock;

osc-ost is a LDLM_EXTENT lock: ost supports EXTENT lock;

Special ldlm_res_id is used:

```
#define FID_SEQ_UPDATE_LOCK (FID_SEQ_START + 3)
#define FID_OID_UPDATE_LOCK (0x0ba771e7)
struct ldlm_res_id barrier_resid = { .name[0] = FID_SEQ_UPDATE_LOCK,
                                     .name[1] = FID_OID_UPDATE_LOCK };
```

Details and Focus for Inspection (2)

mdc/osc checks if it has the DLM lock before sending new request:

- if yes, continue;

- if not, it enqueues such a CR lock.

- Only for update request?

mds/obdfilter get an EX lock before destroying exports and obd cleanup.

- mdc/osc get BAST, and

- cancel all its locks, cache flushed

LDLM_FL_NO_LRU is used

(users may require to get such EX lock on mds/obdfilter via proc.)

Open Request Replay

"Open Handle" should be preserved between fail over.

"Open" request has different packet format in 1.8 and 2.0

Reformat for open request is needed when upgrade

Client is evicted when downgrade from 2.0 to 1.8: No open replay.

1.8 mds server does not understand fid.

Open Issues

When clients loses such lock, it tries to get it immediately, before sending new requests.

When mdt/ost get EX lock, the default timeout value maybe is not long enough: the clients need to flush all cache.

Avoid races: mdt/ost get EX lock, destroy EX lock, ..., destroy the DLM namespace

Server stop processing request when EX lock is held?

Scalability issues?

Resources

Arch page:

[http://arch.lustre.org/index.php?
title=Interoperability_fids_zfs](http://arch.lustre.org/index.php?title=Interoperability_fids_zfs)

[http://arch.lustre.org/index.php?
title=Simplified_Interoperation](http://arch.lustre.org/index.php?title=Simplified_Interoperation)

Bug #:

Bug 11824, for 1.8

Bug 17911, for 2.0



THANK YOU