Recovery and Eviction

Jesse Hanley
Sarp Oral
Neena Imam

January 2017
Overview

• The transaction-based model
• Eviction
• Recovery
• Post-Recovery
Transactions
/* Data stored per client in the last_rcvd file. In le32 order. */

struct lsd_client_data {
    __u8  lcd_uuid[40];       /* client UUID */
    __u64 lcd_last_transno;   /* last completed transaction ID */
    __u64 lcd_last_xid;       /* xid for the last transaction */
    __u32 lcd_last_result;    /* result from last RPC */
    __u32 lcd_last_data;      /* per-op data (disposition for open &c.) */
    /* for MDS_CLOSE requests */
    __u64 lcd_last_close_transno; /* last completed transaction ID */
    __u64 lcd_last_close_xid;  /* xid for the last transaction */
    __u32 lcd_last_close_result; /* result from last RPC */
    __u32 lcd_last_close_data; /* per-op data */
    /* VBR: last versions */
    __u64 lcd_pre_versions[4];
    __u32 lcd_last_epoch;
    /* generation counter of client slot in last_rcvd */
    __u32 lcd_generation;
    __u8  lcd_padding[LR_CLIENT_SIZE - 128];
};
Request Flow

1 - Request
2 - Response
3 - Async Write to Disk
4 - Final Response
Interruptions

• Caused by:
  – Network failure
  – Failing hardware
  – Software issues

• Previously, all issues were treated the same
Impact to the request flow

• Replay:
  – Loss of state in server memory
  – Change wasn’t written to disk

• Resend:
  – Client didn’t receive a reply
  – If change exists already on disk, rebuild the reply message
  – If not, perform the action as if it was the first time
Eviction

- Client invalidates all locks and cached inodes
- Forces flush of cached data
- Generally caused by network communication errors/timeouts
- Retry & Re-establish
Version-Based Recovery

• MDS sends copy of previous and current inode with modifying request
• These are sent during recovery
• Recovery checks these to on-disk versions, rather than locking strictly on transno
Computational Research & Development Programs

Imperative Recovery

[mgs]$
$ lctl get_param mgs.MGS.live.testfs
...

imperative_recovery_state:
  state: full
  nonir_clients: 0
  nidtbl_version: 242
  notify_duration_total: 0.470000
  notify_duration_max: 0.041000
  notify_count: 38
Conclusion

• Improvements to Lustre recovery
  – Version-based recovery includes additional information used to frame the state of the file system around a request
  – Imperative recovery allows MGS to proactively contact clients

• Eviction process

• Recovery process
Resources

• https://build.hpdd.intel.com/job/lustre-manual/lastSuccessfulBuild/artifact/lustre_manual.xhtml


• http://wiki.lustre.org/images/0/00/A_Deep_Dive_into_Lustre_Recovery_Mechanisms.pdf
Acknowledgements

This work was supported by the United States Department of Defense (DoD) and used resources of the Computational Research and Development Programs at Oak Ridge National Laboratory.