Final Report for the MDT-MDT Consistency Subproject of the Lustre File System Checker of the SFS-DEV-001 Contract

Revision History

Date	Revision	Author
03/31/15	First release	R. Henwood

Contents

Executive Summary

This document finalizes the activities undertaken during the Lustre* File System Checker, Sub Project 3.3: MDT-MDT Consistency project within the OpenSFS Lustre Development contract SFS-DEV-001 signed July 30th 2011 and the subsequent modification signed October 10th 2012.

Notable highlights of this project include:

- Demonstrated scalable consistency checking over multiple MDTs tested on the OpenSFS Cluster.
- MDT-MDT functionality was implemented with 16436 new lines of code and was landed on Lustre Master for inclusion into Lustre release 2.7 on March 13th 2015.
- All assets generated for OpenSFS during the project are attached to the OpenSFS wiki: http://wiki.opensfs.org/Contract_SFS-DEV-001

Statement of Work

MDT-MDT Consistency will add concurrent distributed verification and repair for DNE file systems. This will add functionality while the MDT is iterating over its inodes to check that directory entries reference inodes correctly and consistently with each inode's back-pointer to its parent directory. This includes cross-MDT references where the directory entry and inode are located on different MDTs. Incorrect back pointers and orphan inodes will be resolved when detected. This will allow complete checking of DNE file systems.

Subproject 3.1, 3.1.5, 3.2 and this sub-project (3.3) altogether constitute a complete, scalable solution for detecting and automatically correcting corruption on a Lustre file system with DNE. This allows the distributed checking and repair of Lustre inter-server state for while the file system is on-line. LFSCK supports multiple MDTs present on DNE file systems and will run simultaneously on multiple MDTs.

The complete scope statement was agreed on 2014-04-22 and is available at:

http://wiki.opensfs.org/images/c/cd/LFSCK_MDT-MDTConsistency_ScopeStatement.pdf

*Other names and brands maybe the property of others.

Summary of Solution Architecture

With the completion of LFSCK 1.5 in Lustre* software version 2.4, a complete solution for namespace consistency verification for single MDT including the FID-in-dirent and linkEA became available. With Distributed Namespace (DNE) fully enabled a new class of consistency cases are introduced and LFSCK must verify the global namespace consistency across multiple MDTs. In constrast to the single MDT case, there are two main namespace changes related to DNE:

- **Remote Object:** The file (or directory) name entry and the corresponding MDT-object reside on different MDTs. The MDT-x holds the file name entry in the parent directory, and the name entry references the MDT-object with the FID-in-dirent or FID-in-LMA (XATTR_NAME_LMA) EA (of local agent). The MDT-y holds the file's MDT-object, and it points back to the remote name entry and the parent FID with linkEA (XATTR_NAME_LINK).
- **Striped Directory:** With DNE a directory can be striped to multiple MDTs. A striped directory contains one parent MDT-object and 1-N children MDT-objects; the parent MDT-object stores the stripe information as sub-directories and each sub-directory references one MDT-object; the child MDT-object references the parent MDT-object via the ".." entry.

In LFSCK 3, the FID-in-dirent and linkEA verification is extended from single-MDT to cross-MDT cases. In addition, inconsistency check/repair for remote objects and striped directories is included. These additional tests include:

- **Dangling name entry:** The name entry exists, but the related MDT-object does not exist.
- **Orphan MDT-object:** The MDT-object exists, but there is no name entry to reference it.
- **Multiple-referenced name entry:** More than one MDT-object points back to the same name entry, but the name entry only references one of them.
- **Unmatched name entry and MDT-object pairs:** The name entry references the MDT-object which has no linkEA for back-reference or it points back to another name entry that does not exist or does not reference the MDT-object.
- **Unmatched object's type:** The file type in the name entry does not match the type that is claimed by the MDT-object.

- **Invalid nlink count:** The MDT-object's nlink count does not match the name entries (which reference such MDT-object) count.
- **Invalid name hash for striped directory:** The name entry is corrupted and the name hash for striped directory does not match the LMV EA. Verify that the name entry is hashed to the MDT correctly.

In LFSCK phase 1.5, we implemented the functionality of scanning the single MDT device via low layer object-table based iteration and namespace based directory traversal. In this phase, the framework constructed for LFSCK 1.5 will be enhanced and extended to iterate on all MDT devices in parallel.

Acceptance Criteria

The acceptance test will be performed with code running on Lustre master branch. The LFSCK for MDT-MDT consistency will be accepted if the following use cases are demonstrated:

1. Start/stop MDT-MDT consistency check/repair through userspace commands
2. Monitor MDT-MDT consistency check/repair
3. Resume MDT-MDT consistency check/repair from the latest checkpoint
4. Rate control for MDT-MDT consistency check/repair
5. Repair dangling name entry
6. Repair orphan MDT-object
7. Repair multiple-referenced name entry
8. Repair unmatched name entry and MDT-object pairs
9. Repair invalid file type
10. Repair invalid nlink count
11. Repair invalid name hash for striped directory
12. The Lustre system is available during the LFSCK for MDT-MDT consistency check/repair

The complete Solution Architecture including Acceptance Criteria was agreed on 2014-05-29 and is available at:

http://wiki.opensfs.org/images/2/25/LFSCK_MDT-MDTConsistency_SolutionArchitecture.pdf

Summary of High Level Design

MDT-MDT consistency scanning (along with the other phases of LFSCK) is a highly complex implementation task. The complete High Level Design document weighs in at over 11800 words (including feedback comments). That document describes in detail the following considerations:

- Discover MDT-MDT inconsistency
    - First-stage scanning - verify name entries
    - Second-stage scanning - verify multiple-linked files and orphan MDT-objects
- LFSCK tracing
    - Record failed servers during the first-stage scanning
- LFSCK user space control
- LFSCK engines
    - LFSCK master engine
    - LFSCK assistant thread
- Repair MDT-MDT inconsistency
    - Repair dangling name entry
    - Repair unmatched name entry and MDT-object pairs
    - Repair multiple-referenced name entry
    - Repair orphan MDT-object
    - Verify multiple referenced MDT-object - invalid/redundant linkEA entries, invalid nlink count

        - multiple-referenced normal file
        - multiple-referenced directory
    - Repair invalid file type
    - Repair invalid name hash for striped directory
        - The master LMV EA matches the slave LMV EA
        - The directory object has no LMV EA
        - The master LMV EA and the slave LMV EA does not match
- Object visibility changes
    - Export/`lost+found`
- API changes
    - Enhance `dt_index_operations::dio_insert`
    - New `LFSCK_NOTIFY` event - `LE_CREATE_ORPHAN`
    - New `LFSCK_NOTIFY` event - `LE_RESCAN_DIR`
- Race control between MDT-MDT consistency check/repair and other operations
    - Handle orphan MDT-object and cross-MDT create
    - Orphan MDT-object verification and metadata migration
    - Name entry verification and rename
- Recovery process
    - Recovery from the first-stage scanning failures
    - Recovery from the second-stage scanning failures
- Interoperability and Compatibility

The complete High Level Design was agreed on 2013-07-23 and is available at:

http://wiki.opensfs.org/images/f/f2/LFSCK_MDT-MDTConsistency_HighLevelDesign.pdf

Summary of Implementation

LFSCK 3: MDT-OST consistency is implemented in the following patches:

Change #	Subject
10030	LU-4923 lfsck: detailed statistics for namespace LFSCK
10143	LU-4972 lfsck: skip .lustre and children for namespace check
10447	LU-5099 api: transfer object type via dt_insert API
10733	LU-5180 lfsck: linkea for orphan
10751	LU-5223 lmv: build master LMV EA dynamically build via readdir
10765	LU-5223 lod: generate linkEA for shard of striped directory
11275	LU-5208 tests: inject failure on the proper OST
11276	LU-4970 lfsck: flush async updating before exit
11288	LU-5075 test: keep LFSCK fail_loc until recovery completed
11407	LU-5466 lfsck: typo in lfsck_del_target
11590	LU-4970 tests: wait async LFSCK updates to be done
11304	LU-5395 lfsck: misc patch to prevent lfsck hung
11373	LU-5395 lfsck: deadlock between LFSCK and destroy
10986	LU-4788 lfsck: take ldlm lock before modifying visible object
10987	LU-4788 lfsck: verify .lustre/lost+found at the LFSCK start
10602	LU-4788 lfsck: replace cfs_list_t with list_head
10493	LU-4788 lfsck: LFSCK code framework adjustment (1)
10603	LU-4788 lfsck: namespace LFSCK uses assistant thread
10996	LU-5506 lfsck: skip orphan OST-object handling for failed OSTs
11382	LU-5508 osp: RPC adjustment for remote transaction
11485	LU-5509 osd: get PFID from linkEA for remote dir on ldiskfs
11317	LU-4788 lfsck: enable verification for remote object
11486	LU-5511 lfsck: repair unmatched parent-child pairs
11330	LU-5512 lfsck: repair dangling name entry
11383	LU-5513 lfsck: repair multiple referenced name entry
11384	LU-5515 lfsck: repair bad file type in name entry
11390	LU-5516 lfsck: repair the lost name entry
11391	LU-5516 lfsck: repair orphan parent MDT-object
11444	LU-5506 lfsck: skip orphan MDT-object handling for failed MDTs
11516	LU-5517 lfsck: repair invalid nlink count
11536	LU-5518 lfsck: recover orphans from backend lost+found
11714	LU-5519 lfsck: verify striped directory

The Implementation milestone was agreed on 2014-09-18 and is available at:

http://wiki.opensfs.org/images/a/ad/LFSCK_MDT-MDTConsistency_Implementation.pdf

Summary of Demonstration

LFSCK 3: Successfully completed the following demonstration criteria:

- Correctness: coded into sanity-lfsck.sh and sanity-scrub.sh. In addition, standard review tests were passed.
- Measure performance of LFSCK 3 against multiple consistent MDTs.
- Measure performance of LFSCK 3 against multiple MDTs with inconsistencies.
- Measure the impact of LFSCK on small file create performance on multiple MDTs without inconsistencies.

Complete results were agreed on 2015-03-24 and are available at:

http://wiki.opensfs.org/images/5/5a/LFSCK_MDT-MDTConsistency_Demonstration.pdf

Documentation

Documentation was completed as part of issue https://jira.hpdd.intel.com/browse/LUDOC-254 and landed on change http://review.whamcloud.com/12277.