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

Revision History

Date	Revision	Author
07/31/14	Original	R. Henwood
08/07/14	Improvements from OpenSFS team	R. Henwood

Contents

Executive Summary

This document finalizes the activities undertaken during the Lustre* File System Checker, Sub Project 3.2: MDT-OST Consistency project within the OpenSFS Lustre Development contract SFS-DEV-001 signed July 30th 2011.

Notable highlights of this project include:

- Demonstrated scalable consistency checking over multiple MDTs tested on the OpenSFS Cluster.
- MDT-OST functionality was implemented with 12886 new lines of code and was completed and landed on Lustre Master for inclusion into Lustre release 2.6 on July 31st 2014.
- 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-OST consistency implements functionality for distributed verification and repair of the MDT-inode-to-OST-object layout. The LFSCK 2 functionality works in a distributed fashion with the MDT iterating over the inodes to check:

- the file layout (LOV EA) and verify the ostid therein
- the OST objects referenced by the file layout exist
- each OST object has a back reference to the correct MDT inode

Incorrect or missing back pointers on the OST objects are corrected, and missing objects recreated when detected.

The UID and GID of OST objects are also verified to match that of the MDT inode to ensure correct quota allocation. After the MDT iteration is complete, any unreferenced OST objects are linked into the Lustre .lustre/lost+found/ directory.

Subproject 3.1, 3.1.5 and this sub-project (3.2) altogether constitute a complete, scalable replacement of the existing e2fsprogs-based lfsck utility. This allows the distributed checking and repair of Lustre inter-server state for while the file system is

*Other names and brands maybe the property of others.

on-line. LFSCK supports multiple MDTs present on DNE file systems and will run simultaneously on multiple MDTs. MDT to MDT consistency is not supported in this phase of the project.

The complete scope statement was agreed on 2013-05-01 and is available at:

http://wiki.opensfs.org/images/d/d3/LFSCK_MDT-OSTConsistency_ScopeStatement.pdf

Summary of Solution Architecture

Normal files on a Lustre* file system (i.e. non-directory) are composed of one MDT object (the parent) and zero or more OST object(s) (or children). The parent resides on the MDT, and records the file layout information in the Logical Object Volume Extended Attribute (LOV EA) for the children belonging to the file. With the file layout information, a client can locate the specified OST object. To ensure data integrity, each child object on the related OST also records its parent identifier information to indicate to which file the OST object belongs. Under normal operation, the file layout information stored on the parent will be consistent with the parent identifier information stored in its children. With a production system, however, a error/failure condition may arise that can cause the parent-child pointers to become inconsistent. The inconsistency includes the following cases:

1. **Dangling reference.** *MDT-object1* claims *OST-object1* is its child, but on the OST, *OST-object1* does not exist, or is not initialized (and does not recognize *MDT-object1* as its parent).

2. **Unreferenced OST object.** *OST-object1* claims *MDT-object1* is its parent, but on the MDT, *MDT-object1* does not exist, or is not initialized (and does not recognize *OST-object1* as its child).

3. **Mismatched reference.** *MDT-object1* claims *OST-object1* is its child, but *OST-object1* claims that its parent is *MDT-object2*. On the MDT, *MDT-object2* doesn't exist, or it doesn't recognize *OST-object1* as its child.

4. **Multiple references.** *MDT-object1* claims *OST-object1* is its child, but *OST-object1* claims that its parent is *MDT-object2* rather than *MDT-object1*. *MDT-object2* recognizes *OST-object1* as its child.

A further type of inconsistency between an MDT and an OST is concerned with quota. Both MDT objects and OST objects have ownership information (UID and GID) to indicate which user the file/object belongs to. If the owner information for the MDT

object and OST object(s) belonging to the same file are inconsistent, then quota will be inaccurate.

To fix the MDT-OST inconsistencies identified above, a scan of the whole system is needed including both the MDT and OSTs is performed. The existing object-table based iteration, implemented in LFSCK Phase I, scans the whole system. In addition to MDT-OST inconsistencies, all previous LFSCK functionality from phases 1 and 1.5 will execute and correct the file system.

Acceptance Criteria

The criteria that must be met for a successfully acceptance are:

1. The administrator can start and stop MDT-OST consistency check/repair through userspace commands.
2. The administrator can monitor MDT-OST consistency check/repair.
3. The administrator can resume MDT-OST consistency check/repair from the latest checkpoint.
4. The administrator can control the rate of scanning for MDT-OST consistency check/repair.
5. LFSCK will repair file for which the parent has a dangling reference.
6. LFSCK will repair an unreferenced OST object.
7. LFSCK will repair an unmatched referrenced MDT object and OST object pair.
8. LFSCK will repair a repeat referenced OST object.
9. LFSCK will repair inconsistent file owner information.
10. The administrator can upgrade an OST from Lustre 1.8.
11. The Lustre system is available during the LFSCK for MDT-OST consistency check/repair.

The complete Solution Architecture including Acceptance Criteria was agreed on 2013-06-05 and is available at:

http://wiki.opensfs.org/images/e/ea/LFSCK_MDT-OSTConsistency_SolutionArchitecture.pdf

Summary of High Level Design

MDT-OST 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 9000 words. That document describes in detail the following considerations:

- Identifying inconsistencies efficiently in a distributed environment.
- Tracing LFSCK during operation.
- Controlling and monitoring LFSCK from user space.
- Design of the independent LFSCK check/repair engines.
- Strategy to repair different inconsistencies in a distributed environment.
- Notification of events to the changelog.
- Recovery in the event of failure of any part of the system.
- Changes needed to the wire protocol to support MDT-OST consistency check/repair.
- Changes to the OSD API.
- Control and resolution of race conditions with LFSCK and other operations.
- On-disk layout changes.
- Interoperability and compatibility with other features.

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

http://wiki.opensfs.org/images/b/b9/LFSCK_MDT-OSTConsistency_HighLevelDesign.pdf

Summary of Implementation

LFSCK 2: MDT-OST consistency is implemented in the following patches:

ID	Description
591f15e	LU-4106 scrub: Trigger OI scrub properly
d628a95	LU-3951 lfsck: OST object inconsistency self detect/repair
b3e6eda	LU-3951 lfsck: LWP connection from OST-x to MDT-y
bf15de9	LU-3950 lfsck: control LFSCK on all devices via single command
80054e6	LU-3594 lfsck: repair inconsistent OST object owner
cd21da7	LU-3593 lfsck: repair inconsistent layout EA
aa87bcd	LU-3592 lfsck: repair multiple referenced OST object
74c1059	LU-3591 lfsck: repair unmatched MDT-OST objects pairs
aa1e2e7	LU-3590 lfsck: repair MDT object with dangling reference
f792da4	LU-3569 ofd: packing ost_idx in all IDIF
2a271b4	LU-3336 lfsck: create new MDT object or exchange OST objects
ac78f55	LU-3336 lfsck: recreate the lost MDT object
416839c	LU-3336 lfsck: namespace visible lost+found directory
c61c112	LU-3336 lfsck: regenerate lost layout EA
56fdc3c	LU-3336 lfsck: orphan OST objects iteration (2)
5b4f73b	LU-3336 lfsck: orphan OST objects iteration (1)
4425fa1	LU-3336 lfsck: use rbtree to record OST object accessing
3e09d63	LU-3335 osd: use local transaction directly inside OSD
cc581f3	LU-1267 lfsck: enhance API for MDT-OST consistency
7a998f3	LU-1267 lfsck: enhance RPCs (2) for MDT-OST consistency
a956e98	LU-1267 lfsck: enhance RPCs (1) for MDT-OST consistency
edb1bd9	LU-1267 lfsck: framework (3) for MDT-OST consistency
f946d82	LU-1267 lfsck: framework (2) for MDT-OST consistency
141a375	LU-1267 lfsck: rebuild LAST_ID
ff36f471	LU-1267 lfsck: framework (1) for MDT-OST consistency

The Implementation milestone was agreed on 2013-09-27 and is available at:

http://wiki.opensfs.org/images/e/ee/LFSCK_MDT-OSTConsistency_Implementation.pdf

Summary of Demonstration

LFSCK 2: Successfully completed the following demonstration criteria:

- Correctness: coded into sanity-lfsck.sh and sanity-scrub.sh
- Standard review tests: sanity, sanityn, replay-single, conf-sanity, recovery-small, replay-ost-single, insanity, sanity-quota, sanity-sec, lustre-rsync-test, lnet-selftest, and mmp
- Measure performance of LFSCK 2 against a single MDT device without inconsistencies
- Measure performance of LFSCK 2 against a single MDT device with inconsistencies
- Impact of LFSCK 2 on small file create performance on a single MDT with inconsistencies
- Measure performance of LFSCK 2 against multiple MDT devices during inconsistency repair.

Complete results were agreed on 2013-11-13 and are available at:

http://wiki.opensfs.org/images/0/08/LFSCK_MDT-OSTConsistency_Demonstration.pdf

Documentation

Documentation was completed as part of issue http://review.whamcloud.com/9068 and landed on change https://jira.hpdd.intel.com/browse/LUDOC-155.

In addition, a design document for LFSCK 2 has been included in the Lustre source code tree in the 'Documentation' folder.