

Scope statement

LFSCK 1.5: FID-in-Dirent and LinkEA Scope Statment

Introduction

The following scope statement applies to LFSCK 1.5: FID-in-Dirent and LinkEA Consistency Scope Statement project within the SFS-DEV-001 contract and associated modifications.

Problem Statement

Over the lifetime of an active filesystem, FID-in-Dirent and linkEA entries may become inconsistent or invalid as the result of:

- non-graceful shutdown.
- restoration from MDT file-level backup.
- upgrade from Lustre-1.8.

An MDT upgraded from Lustre-1.8 has neither FID-in-Dirent nor linkEA is available. Without FID-in-Dirent records, Lustre-2.x will not achieve maximum performance of read-dir on the MDT.

LFSCK 1.5 will implement functionality to verify and rebuild FID-in-Dirent and linkEA. This will be achieved by additional functionality executed while the MDT is iterating over the inodes (see Subproject 3.1: inode iterator and OI Scrub) to verify the FID-in-Dirent name entry is consistent with the FID in the inode LMA. If an inconsistency is discovered it will be repaired or rebuild if FID-in-Dirent name entry is absent. The name entry is also verified that it correctly references the inode linkEA and the inode linkEA points back to the valid name entry. The unmatched or redundant inode linkEA will be removed, and the correct or missed inode linkEA will be added.

In addition, the FID-in-Dirent process will add IGIF FIDs to the dirent for upgraded 1.8 file systems. This additional step will avoid the need to look up the inode when doing readdir.

Project Goals

1. FID-in-DIR and linkEA is available on the MDT restored from file-level backup.
2. FID-in-DIR and linkEA is available on the MDT upgraded from Lustre-1.8 format device.
3. Characterize the performance of LFSCK without filesystem load.
4. Characterize the performance of LFSCK with filesystem load.
5. Land code to Lustre Community Release master branch.

In-Scope

- Kernel-space FID-in-Dirent and linkEA checking and repairing on single MDT.
- User-space tools to control LFSCK for for FID-in-Dirent and linkEA consistency.
- Administrative documentation in the form of a man page and update to Intel Lustre-2.x manual.

Out of Scope

- The LFSCK for FID-in-Dirent and linkEA will be implemented against the inode iterator introduced in LFSCK phase I (Subproject 3.1: inode iterator and OI Scrub). Only ldiskfs OSDs will be implemented, tested and landed for this phase.
- LFSCK for FID-in-Dirent and linkEA consistency will act only on a single MDT at this phase.
- Detection and resolution of internal file-system inconsistencies is not within LFSCK scope.

Project Constraints

- Fan Yong is the only engineer with the correct expertise available for this work.

Key Deliverables

- Signed Milestone documents for project phases:
 - Solution Architecture.
 - Design Document.
 - Implementation & Test.
 - Acceptance Testing (OpenSFS executed).
- Test Plan.
- Land LFSCK code (including new test cases) for FID-in-Dirent and linkEA consistency to Lustre-2.x.

Key Milestones

Fan Yong is available now.

- Scope Statement: 2012-10-12
- Solution Architecture: 2012-10-26
- Design Document: 2012-11-09
- Implementation: 2012-12
- Acceptance Testing: 2013-01
- Demonstration: 2013-01

Glossary

- FID-in-Dirent
In Lustre-2.x, when a file is created its FID will be stored as part of the name entry in its parent directory. This is known as FID-in-Directory-Entry or 'FID-in-Dirent.' With the FID-in-Dirent, the readdir on MDT can find the file FID in the directory page directly. Without FID-in-Dirent, the directory page must be identified by reading from inode LMA. As a result, traversing the directory (such as "ls") is accelerated with FID-in-Dirent.
- linkEA
In Lustre-2.x, when a file is created or hard linked, the name together with the parent directory FID are stored as extended attributes. This is known as link-extended-attributes, or linkEA. With the linkEA the given FID can be parsed back to a full path from the root to the target file. This is useful for Changelog based applications, such as lustre_rsync.