Lustre Feature Test Plan for *Lazy Size on MDS*

Revision 0.3
December 12, 2018

Revision History

The following is a chronological history of changes made to this document.

Revision	Date	Reason for change	Author
0.1	September 11, 2018	Initial Version	James Nunez
0.2	September 27, 2018	Add set up steps for LSOM, add requirement of changelog user to llsom_sync utility.	James Nunez
0.3	December 12, 2018	Incorporated input from Andreas Dilger, Li Xi and Qian Yingjin. Clarify that LSOM will not improve performance until integrated with stat().	Andreas Dilger/James Nunez

Contents

Revision History 2

Introduction 4

Documentation 4

Feature Installation and Set-up 4

Regression Tests 5

Manual Testing 5

Interoperability 5

Performance Testing 5

User Interface 5

lfs getsom 5

llsom_sync 6

API 7

Feature Interaction 7

What Not to Test 7

Results 7

Introduction

In the Lustre file system, the Metadata Servers (MDSs) store the ctime, mtime, owner, and other file attributes. The Object Storage Servers (OSSs) store the size and number of blocks used for each file. To obtain the correct file size, the client must contact each Object Storage Target (OST) that the file is stores across, which means multiple RPCs to get the size and blocks for a file when a file is striped over multiple OSTs. The Lazy Size on MDS (LSOM) feature stores the file size on the MDS and, avoids the need to fetch the file size from the OST(s) in some cases where the application understands that the size may not be accurate. Lazy means there is no guarantee of the accuracy of the attributes stored on the MDS.

Since many Lustre installations use SSD for the Metadata Target (MDT) storage, the motivation for this work is to speed up the time it takes to get the size of a file from the Lustre file system by storing that data on the MDTs. We expect this feature to be initially used by Lustre policy engines like Robinhood and DDN's Lustre Integrated Policy Engine (LiPE) that scan the filesystem and make decisions based on broad size categories, and do not depend on a totally accurate file size. Future improvements will allow the LSOM data to be accessed by tools such as `lfs find`.

Documentation

The Lazy Size on MDS JIRA ticket, LU-9538, has detail and discussion on the feature.

A man page for the `llsom_sync` utility is included with Lustre 2.12. In addition, slides for a LSOM talk from the 2018 Lustre Users Group Meeting are at http://cdn.opensfs.org/wp-content/uploads/2018/04/Xi-Lazy_Size_on_MDS_DDN.pdf .

Currently, there is no documentation for this feature in the Lustre manual, but we have an open JIRA ticket, LUDOC-402. The lfs man page needs to be updated to include the new "getsom" flag.

Feature Installation and Set-up

The Lazy Size on MDS is always enabled and, thus, you do not have to do anything to start the feature. In order to use and get correct LSOM data, in the current implementation, the client must set `llite.*.xattr_cache` to zero. If `xattr_cache` is set to one, the client will cache the xattrs locally and not fetch them from the MDS. Even if a client truncates/closes a file that changes the LSOM xattr, the cached LSOM xattr data on the client may be stale. For testing purposes, it is enough just to cancel the MDC locks on the client, instead of disabling the xattr cache completely.

If the `llsom_sync` utility will be used, a changelog user must be registered.

Regression Tests

Tests have been added to existing Lustre test suites to cover this feature during autotesting. The following test suites have tests added to cover the LSOM feature:

- sanity test 806 – verifies lazy size on MDS
- sanity test 807 – verifies the LSOM `llsom_sync` utility
- sanity test 808 – verifies trusted.som xattr not logged in Changelogs
- sanity test 809 - verifies no LSOM xattr is stored for DoM files

Manual Testing

No manual testing is needed to test this feature.

Interoperability

The LSOM feature is not compatible with any Lustre versions prior to 2.12.0 and no interoperability testing will be done for this feature.

Performance Testing

Currently, stat does not work with LSOM so 'ls –l' and mdtest will not see an improvement in performance; LU-11554 and LU-11367. Once this is complete, we should run performance testing to record the speed up that LSOM gives on stat and run mdtest to evaluate LSOM performance impact.

User Interface

The lfs command-line interface was extended to get data directly from the MDS. In addition, a new utility, `llsom_sync`, was added to sync LSOM xattrs.

lfs getsom

The `lfs getsom` command lists file attributes that are stored on the MDS. `lfs getsom` is called with the full path and file name for a file on the Lustre file system. If no flags are used, then all file attributes stored on the MDS will be shown.

Usage: lfs getsom [-s] [-b] [-f] <path>

Flags for lfs getsom are:

Flag	Description
-s	Only show the size value of the SOM data for a given file. This is an optional flag.
-b	Only show the blocks value of the SOM data for a given file. This is an optional flag.
-f	Only show the flag value of the SOM data for a given file. This is an optional flag. Valid flags are:

	SOM_FL_UNKNOWN = 0x0000 - Unknown or no SoM data, must get size from OSTs. SOM_FL_STRICT = 0x0001 - Known strictly correct, FLR or DoM file (SoM guaranteed). SOM_FL_STALE = 0x0002 - Known stale - was right at some point in the past, but it is known (or likely) to be incorrect now (e.g. opened for write). SOM_FL_LAZY = 0x0004 - Approximate, may never have been strictly correct, need to sync SOM data to achieve eventual consistency.

llsom_sync

The `llsom_sync` command allows the user to sync the file attributes on the MDS to sync with the valid/up-to-date data on the OSTs. `llsom_sync` is called on the client with the client mount point for the Lustre file system. `llsom_sync` uses Lustre MDS changelogs and, thus, a changelog user must be registered to use this utility.

Usage: llsom_sync --mdt|-m <mdt> --user|-u <user id> [--daemonize|-d] [--verbose|-v] [--interval|-i] [--min-age|-a] [--max-cache|-c] [--sync|-s] <lustre_mount_point>

Flags for llsom_sync are:

Flag	Description
--mdt|-m <mdt>	The metadata device which need to be synced the LSOM xattr of files. A changelog user must be registered for this device. Required flag.
--user|-u <user id>	The changelog user id for the above MDT device. Required flag.
[--daemonize|-d]	Optional flag to “daemonize” the program. In daemon mode, the utility will scan, process the changelog records and sync the LSOM xattr for files periodically.
[--verbose|-v]	Optional flag to produce verbose output.
[--interval|-i]	Optional flag for the time interval to scan the Lustre changelog and process the log record in daemon mode.
[--min-age|-a]	Optional flag for the time that llsom_sync tool will not try to sync the SOM data for any files

	closed less than this many seconds old. The default min-age value is 600s (10 minutes).
[--max-cache|-c]	Optional flag for the total memory used for the FID cache which can be with a suffix [KkGgMm]. The default max-cache value is 256MB. For the parameter value < 100, it is taken as the percentage of total memory size used for the FID cache instead of the cache size.
[--sync|-s]	Optional flag to sync file data to make the dirty data out of cache to ensure the blocks count is correct when update the file LSOM xattr. This option could hurt server performance significantly if thousands of fsync requests are sent.

API

The LSOM extended attributes are accessible through the `llsom_sync` utility and by fgetxattr directly. There is no API to access the LSOM attributes.

Feature Interaction

LSOM should work with all file types and not with directories.

What Not to Test

How to get correct ctime/mtime via LSOM is still a problem and is not implemented. See the comments of LU-9538 for details.
Performance, as measured by mdtest or any benchmarks that use stat(), will not see any performance gain until the work for LU-11554 and/or LU-11367 is complete.

Results

Early results on the impact of LSOM on stat can be found in the LSOM talk given at LUG 2018 at http://cdn.opensfs.org/wp-content/uploads/2018/04/Xi-Lazy_Size_on_MDS_DDN.pdf . Other results will be included here when they are available as appropriate.