DNE2 Solution Architecture

> **Signed-off**
> This document was agreed and signed-off by the PAC on 2013-06-13.

Introduction

DNE Phase 1: Remote Directories made multiple metadata servers a reality on a Lustre* file system. This first phase included some limitations, namely:

1. The namespace can only be distributed to other MDTs by creating sub directory, i.e. a directory on a different MDT (aka remote directory). Typically an administrator may create remote directories for individual users, then those users will do operations in their directory serviced by a given MDT. A typical user will not benefit from simultaneously using multiple MDTs on their own jobs.
2. Except create/unlink remote directory, other cross-MDT operations return -EXDEV. In addition, cross-MDT operation are synchronous to simplify recovery.
3. All name entries in one directory can only exist on a single MDT. Single directory performance for operations like open/create files under one shared directory is the same as single MDT file system.
4. Moving a file onto a remote directory currently requires the file to be copied within the namespace. This is currently inefficient as redundant traffic between OSTs is generated by the copy operation.

DNE Phase 2: Striped Directories address these limitations and enables multiple MDTs on multiple MDS nodes serving a single directory.

Solution requirements

DNE phase II will address the current limitations with the following enhancements:

1. All metadata operations can be across-MDT. Currently only creating/unlinking remote directories can be across-MDT. Users without administrative privilege can execute an across-MDT operation.
2. Across-MDT operation will not be synchronous, and recovery after failure will be supported.
3. Single directory performance will be faster with multiple MDTs.
4. A user can migrate files/directories from one MDT to another MDT without moving data on OSTs.
5. Single MDT file system operations operate as before.

Use Case

Migrate files and directories to the new MDT

A new MDT is added to the file system. A user moves some files to this new MDT efficiently:

1. lfs mv -i MDTn_index dir
2. All of files and sub-directories will be moved to MDTn without needing to move data between OSTs.

Asynchronous cross-MDT operation

A user has a job with a number of cross-MDT operations. With DNE phase II all cross-MDT operations will be asynchronous:

1. Client send metadata requests to the master MDT.
2. The master MDT distribute the request to other remote MDTs, which will handle the requests asynchronously.
3. If there is a MDT failure during this process, the asynchronous mechanism will make the whole distributed operation atomic.

Single directory performance

A users job has a large number of threads. All threads open/create millions of files in a single shared directory. In DNE phase II, the user can:

1. Add a few more MDTs to the file system.

2. Create a striped directory for the job, and the directory will be striped to different MDTs.
3. Open/create work load will be distributed evenly among these MDTs, and the performance will be improved.

Solution proposal

In DNE phase I all cross-MDT operations are synchronous to simplify recovery between MDTs. In DNE phase II, across-MDT operations will be asynchronous and recovery from failures will be required. A redo log (implemented by llog) will be employed to provide recovery from failures:

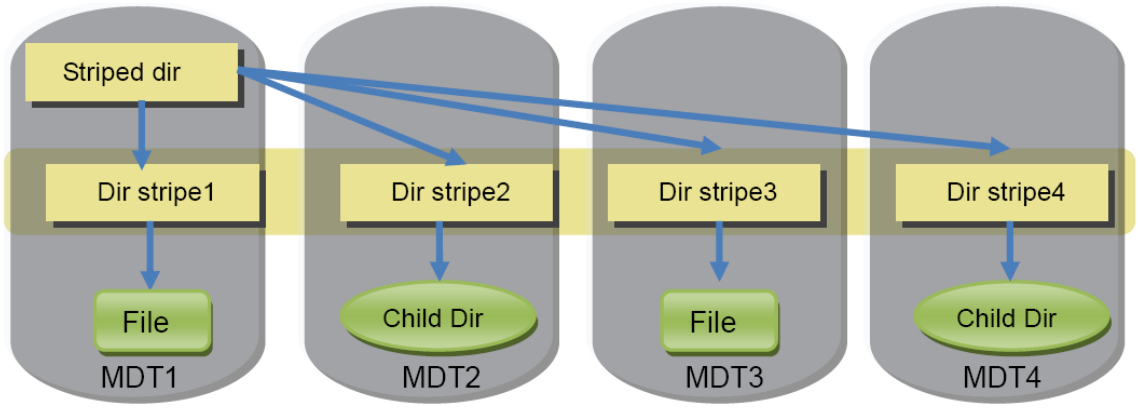
1. A client sends request to the master MDT.
2. Master MDT decomposes the request into several updates and redistributes these updates to remote MDTs.
3. Both master MDT and remote MDT will write these updates into the update log, which will be committed in the same transaction with the update.
4. When the updates are committed on all of MDTs the correspondent update log will be cancelled.

Recovery

For recovery between MDSs, all updates for one operation will be re-distributed on all MDTs and written to the log with the same transaction of operation. Once updates have been committed on one MDT there will be a complete set of updates on the MDT. When one MDT fails, it will notify all other MDTs on reboot to check the correspondent tier of the update log, i.e. the log records which includes the updates on the failover MDT. Update records are then sent to the failover MDT. The failover MDT will check the updates and replay the updates. Finally, the client will send replay request to the failover MDT, which will replay those uncommitted request. Replay from clients will behave as it does currently for a single MDT recovery.

Striped directory

Single directory performance improvements will be resolved by striped directory. Similar to file striping, the directory will be stripped over several MDTs.



Striped Directory

The figure illustrates a directory striped over 4 MDTs. Each stripe occupies a hash space. Suppose the entire hash space is [0, MAX_HASH]: Dir stripe1 will hold name entries with hash [0, MAX_HASH/4 -1], Dir stripe 2 will hold name entries with [MAX_HASH/4, MAX_HASH/4 * 2 - 1], Dir stripe3 will hold name entries with hash [MAX_HASH/4*2, MAX_HASH/4*3 -1], Dir stripe4 will hold name entries that has [MAX_HASH/4*3, MAX_HASH]. During name insertion or lookup, a client will calculate the hash value by name and select the appropriate MDT to service the request to the stripe based on the hash value. During normal operations, the MDTs handle requests independently: i.e. there is no communication between MDTs. This design will improve the performance of single directory operations.

Test plan

Functional test

test	modification	pass condition
sanity	1. add sanity test to check cross-MDT operation. 2. add sanity test to check striped directory create/unlink. 3. Add sanity test to check migration tool. It should include migrate striped directory. 4. Check each test, if there are cross directory operation, make two directories on different MDT.	Those function works as expected.
conf-sanity	add upgrade test from 2.4 to 2.5 in test_32.	The disk-image created by phaseI can be setup with DNE phase II(lustre 2.6), and run a few simple tests, it should pass.
racer	create directories on every MDTs and do cross-MDT operation inside racer.	the test should run for at least 10 mins, and no LBUG or deadlock.
meta-performance	Add striped directory test for metadata performance test.(mdsrate-xxx.sh)	the rate(open/sec, unlink/sec etc) should be better than doing test on single directory
recovery-small	Drop reply in different phases for different cross-MDT operation. (rename, link, setdirstripe etc) 1. a. Master MDT drops reply to client. client will resend request. b. slave MDT drops reply to Master MDT. Master MDT will resend request. c. Slave MDT drops reply of llog cancel to Master MDT. Master MDT will resend llog cancel request.	No application failure in those failover test.
replay-single	Fail single or multiple MDTs in different cross-MDT operation, like rename, setdirstripe, link etc	No application failure in those failover test.
replay-dual	Fail single or Multiple MDTs in different cross-MDT operation, like rename, setdirstripe, link etc	No application failure in those failover test.
sanity-quota	Do cross-MDT operation then check the inode quota.	Inode quota count should be changed as expected after cross-MDT operation.
lustre-rsync-test	Add test to rsync striped directory/remote directory to lustre/non-lustre system.	After lustre-rsync, the target and the source should be same.

Recovery test

1. Setup lustre with 8 clients, 4MDS(2MDTs per MDS), 4 OST.
2. Run different tests(dbench, iozone, dd) on each client. Half of the tests must be run under striped directory. All of test should include cross-MDT operation.

3. Choose 1 MDS to fail every 30 mins, and keep running test for 24 hours, and there should not be any test failure during the test.

Compatibility test

1. Setup lustre with old clients (Lustre 2.4) and new MDSs (Lustre master).
    a. Typical test suites included in Lustre 2.4 will pass.
    b. Add Lustre master client, then create a striped directory. A Lustre 2.4 client cannot access the directory.
2. Setup lustre with new clients (Lustre master) and old MDSs (Lustre 2.4)
    a. Typical test suites included in Lustre 2.4 will pass.
    b. Attempt to create a striped directory will return -ENOTSUPP.
    c. Attempt to access a striped directory will return -ENOTSUPP.
    d. Attempt to do an cross-MDT operation (other than create/unlink remote directory) will return -EXDEV.

Performance Test

1. Measure multiple clients writing to a single directory with a single MDT in the previous release.
2. Measure multiple clients writing to a single directory with 1, 2, 3, and 4 stripes with four MDTs.

Acceptance Criteria

All of tests will pass
