

**Milestone Completion for the  
LFSCK 3 MDT-MDT Consistency on the  
Lustre\* software FSCK Project of the  
SFS-DEV-001 contract.**

Revision History

<b>Date</b>	<b>Revision</b>	<b>Author</b>
2014-09-09	Original	R. Henwood

## Contents

Introduction.....	3
Subproject Description.....	3
Milestone Completion Criteria.....	3
Location of Completed Solution.....	3
Functional tests.....	5
Demonstration of LFSCK 3 functionality.....	5
Conclusion.....	6
Appendix A: functional test results from 2014-09-06.....	7

## Introduction

The following milestone completion document applies to Subproject 3.3 – LFSCCK 3: MDT-MDT Consistency of the Lustre\* FSCK within Amendment No. 1 on the OpenSFS Lustre Development contract SFS-DEV-001 agreed October 10, 2012.

## Subproject Description

Per the contract, Implementation milestone is described as follows:

*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.*

## Milestone Completion Criteria

Per the contract, Implementation milestone is described as follows:

*Contractor shall complete implementation and unit testing for the approved solution. Contractor shall regularly report feature development progress including progress metrics at project meetings and engineers shall share interim unit testing results as they are available. OpenSFS at its discretion may request a code review. Completion of the implementation phase shall occur when the agreed to solution has been completed up to and including unit testing and this functionality can be demonstrated on a test cluster. Code Reviews shall include:*

- a. *Discussion led by Contractor engineer providing an overview of Lustre source code changes*
- b. *Review of any new unit test cases that were developed to test changes*

## Location of Completed Solution

The agreed solution has been completed and is recorded in the following patches:

---

\* Other names and brands maybe the property of others.

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

## Functional tests

New functional tests to automatically verify the acceptance criteria agreed in the [LFSCCK3 Solution Architecture](#) are available. The tests are contained in modifications to the files:

lustre/tests/sanity-lfsck.sh

lustre/tests/test-framework.sh

NOTE: In addition to new tests to verify new functionality, LFSCCK uses existing tests to verify rate control and start/stop/monitor of LFSCCK.

## Demonstration of LFSCCK 3 functionality.

Functional testing was completed on 2014-09-02. The detailed Use Cases demonstrated are recorded in Appendix A. Section 2 of the [LFSCCK 3 Solution Architecture](#). The acceptance tests include:

Acceptance test	Corresponding code test from Appendix A
2.1 Start/stop MDT-OST consistency check/repair through userspace commands	Present in all tests
2.2 Monitor MDT-OST consistency check/repair	Present in all tests
2.3 Resume MDT-OST consistency check/repair from the latest checkpoint	test 6a: LFSCCK resumes from last checkpoint test 6b: LFSCCK resumes from last checkpoint test 7a: LFSCCK resumes from last checkpoint test 7b: LFSCCK resumes from last checkpoint
2.4 Rate control for MDT-OST consistency check/repair	test 9a: LFSCCK speed control test 9b: LFSCCK speed control
2.5 Repair dangling name entry	test 23a: LFSCCK repair dangling name entry test 23b: LFSCCK repair dangling name entry test 23c: LFSCCK repair dangling name entry
2.6 Repair orphan MDT-object	test 26a: Repair orphan MDT-object test 26b: Repair orphan MDT-object test 27a: Repair orphan MDT-object test 27b: Repair orphan MDT-object
2.7 Repair multiple-referenced name entry	Test 24: Repair multiple-referenced name

	entry
2.8 Repair unmatched name entry and MDT-object pairs	test 22a: Repair unmatched name entry and MDT-object pairs. test 22b: Repair unmatched name entry and MDT-object pairs.
2.9 Repair invalid file type	test 25: Repair invalid file type.
2.10 Repair invalid nlink count	Test 29a: Repair invalid nlink count. Test 29b: Repair invalid nlink count. Test 29c: Repair invalid nlink count.
2.11 Repair invalid name hash for striped directory	test 31a: Repair invalid name hash for striped directory test 31b: Repair invalid name hash for striped directory test 31c: Repair invalid name hash for striped directory test 31d: Repair invalid name hash for striped directory test 31e: Repair invalid name hash for striped directory test 31f: Repair invalid name hash for striped directory test 31g: Repair invalid name hash for striped directory test 31h: Repair invalid name hash for striped directory
2.12 Lustre system is available during MDT-MDT consistency checking.	Test 10: system is available during MDT-MDT consistency checking

## Conclusion

Implementation has been completed according to the agreed criteria.

## Appendix A: functional test results from 2014-09-06

```
== sanity-lfsck test 6a: LFSCK resumes from last checkpoint (1) == 06:17:08 (1409984228)
preparing... 5 * 5 files will be created Sat Sep 6 06:17:09 UTC 2014.
total: 5 creates in 0.01 seconds: 657.89 creates/second
total: 5 creates in 0.01 seconds: 722.13 creates/second
total: 5 creates in 0.01 seconds: 660.31 creates/second
prepared Sat Sep 6 06:17:10 UTC 2014.
CMD: shadow-7vm8 /usr/sbin/lctl set_param fail_val=1 fail_loc=0x1600
fail_val=1
fail_loc=0x1600
CMD: shadow-7vm8 /usr/sbin/lctl lfsck_start -M lustre-MDT0000 -t namespace -r
Started LFSCK on the device lustre-MDT0000: scrub namespace
CMD: shadow-7vm8 /usr/sbin/lctl get_param -n mdd.lustre-MDT0000.lfsck_namespace
CMD: shadow-7vm8 /usr/sbin/lctl set_param fail_loc=0x80001608
fail_loc=0x80001608
CMD: shadow-7vm8 /usr/sbin/lctl get_param -n mdd.lustre-MDT0000.lfsck_namespace |
      awk '/^status/ { print \$2 }'
CMD: shadow-7vm8 /usr/sbin/lctl get_param -n mdd.lustre-MDT0000.lfsck_namespace |
      awk '/^status/ { print \$2 }'
Waiting 6 secs for update
CMD: shadow-7vm8 /usr/sbin/lctl get_param -n mdd.lustre-MDT0000.lfsck_namespace |
      awk '/^status/ { print \$2 }'
CMD: shadow-7vm8 /usr/sbin/lctl get_param -n mdd.lustre-MDT0000.lfsck_namespace
CMD: shadow-7vm8 /usr/sbin/lctl set_param fail_val=1 fail_loc=0x1600
fail_val=1
fail_loc=0x1600
CMD: shadow-7vm8 /usr/sbin/lctl lfsck_start -M lustre-MDT0000 -t namespace
Started LFSCK on the device lustre-MDT0000: scrub namespace
CMD: shadow-7vm8 /usr/sbin/lctl get_param -n mdd.lustre-MDT0000.lfsck_namespace
CMD: shadow-7vm8 /usr/sbin/lctl get_param -n mdd.lustre-MDT0000.lfsck_namespace
CMD: shadow-7vm8 /usr/sbin/lctl set_param fail_loc=0 fail_val=0
fail_loc=0
fail_val=0
CMD: shadow-7vm8 /usr/sbin/lctl get_param -n mdd.lustre-MDT0000.lfsck_namespace |
      awk '/^status/ { print \$2 }'
CMD: shadow-7vm8 /usr/sbin/lctl get_param -n mdd.lustre-MDT0000.lfsck_namespace |
      awk '/^status/ { print \$2 }'
Resetting fail_loc on all nodes...CMD: shadow-7vm4,shadow-7vm6,shadow-7vm7,shadow-
7vm8,shadow-7vm9.shadow.whamcloud.com lctl set_param -n fail_loc=0 fail_val=0
2>/dev/null || true
done.
CMD: shadow-7vm4,shadow-7vm6,shadow-7vm7,shadow-7vm8 rc=\$([ -f /proc/sys/lnet/catastrophe ]
&&
      echo \$(< /proc/sys/lnet/catastrophe) || echo 0);
      if [ \$rc -ne 0 ]; then echo \$(hostname): \$rc; fi
      exit \$rc
PASS 6a (10s)
```

```
== sanity-lfsck test 6b: LFSCK resumes from last checkpoint (2) == 06:17:19 (1409984239)
preparing... 5 * 5 files will be created Sat Sep 6 06:17:19 UTC 2014.
total: 5 creates in 0.01 seconds: 625.79 creates/second
total: 5 creates in 0.01 seconds: 612.52 creates/second
total: 5 creates in 0.01 seconds: 652.91 creates/second
prepared Sat Sep 6 06:17:20 UTC 2014.
CMD: shadow-7vm8 /usr/sbin/lctl set_param fail_val=1 fail_loc=0x1601
fail_val=1
fail_loc=0x1601
CMD: shadow-7vm8 /usr/sbin/lctl lfsck_start -M lustre-MDT0000 -t namespace -r
Started LFSCK on the device lustre-MDT0000: scrub namespace
CMD: shadow-7vm8 /usr/sbin/lctl get_param -n mdd.lustre-MDT0000.lfsck_namespace
CMD: shadow-7vm8 /usr/sbin/lctl set_param fail_loc=0x80001609
fail_loc=0x80001609
CMD: shadow-7vm8 /usr/sbin/lctl get_param -n mdd.lustre-MDT0000.lfsck_namespace |
      awk '/^status/ { print \$2 }'
CMD: shadow-7vm8 /usr/sbin/lctl get_param -n mdd.lustre-MDT0000.lfsck_namespace |
      awk '/^status/ { print \$2 }'
Waiting 6 secs for update
CMD: shadow-7vm8 /usr/sbin/lctl get_param -n mdd.lustre-MDT0000.lfsck_namespace |
      awk '/^status/ { print \$2 }'
CMD: shadow-7vm8 /usr/sbin/lctl get_param -n mdd.lustre-MDT0000.lfsck_namespace
CMD: shadow-7vm8 /usr/sbin/lctl get_param -n mdd.lustre-MDT0000.lfsck_namespace
CMD: shadow-7vm8 /usr/sbin/lctl set_param fail_val=1 fail_loc=0x1601
fail_val=1
fail_loc=0x1601
CMD: shadow-7vm8 /usr/sbin/lctl lfsck_start -M lustre-MDT0000 -t namespace
Started LFSCK on the device lustre-MDT0000: scrub namespace
CMD: shadow-7vm8 /usr/sbin/lctl get_param -n mdd.lustre-MDT0000.lfsck_namespace
CMD: shadow-7vm8 /usr/sbin/lctl get_param -n mdd.lustre-MDT0000.lfsck_namespace
CMD: shadow-7vm8 /usr/sbin/lctl get_param -n mdd.lustre-MDT0000.lfsck_namespace
CMD: shadow-7vm8 /usr/sbin/lctl set_param fail_loc=0 fail_val=0
fail_loc=0
fail_val=0
CMD: shadow-7vm8 /usr/sbin/lctl get_param -n mdd.lustre-MDT0000.lfsck_namespace |
      awk '/^status/ { print \$2 }'
CMD: shadow-7vm8 /usr/sbin/lctl get_param -n mdd.lustre-MDT0000.lfsck_namespace |
      awk '/^status/ { print \$2 }'
Waiting 32 secs for update
CMD: shadow-7vm8 /usr/sbin/lctl get_param -n mdd.lustre-MDT0000.lfsck_namespace |
      awk '/^status/ { print \$2 }'
Resetting fail_loc on all nodes...CMD: shadow-7vm4,shadow-7vm6,shadow-7vm7,shadow-
7vm8,shadow-7vm9.shadow.whamcloud.com lctl set_param -n fail_loc=0 fail_val=0
2>/dev/null || true
done.
CMD: shadow-7vm4,shadow-7vm6,shadow-7vm7,shadow-7vm8 rc=\$([ -f /proc/sys/lnet/catastrophe ]
&&
      echo \$(< /proc/sys/lnet/catastrophe) || echo 0);
```



```
if [ \${rc} -ne 0 ]; then echo \$(hostname): \${rc}; fi
exit \${rc}
```

PASS 6b (15s)

== sanity-lfsck test 7a: non-stopped LFSCK should auto restarts after MDS remount (1) ==  
06:17:33 (1409984253)

preparing... 5 \* 5 files will be created Sat Sep 6 06:17:34 UTC 2014.

total: 5 creates in 0.01 seconds: 659.81 creates/second

total: 5 creates in 0.01 seconds: 640.27 creates/second

total: 5 creates in 0.01 seconds: 755.62 creates/second

prepared Sat Sep 6 06:17:35 UTC 2014.

10.1.4.75@tcp:/lustre /mnt/lustre lustre rw,flock,user\_xattr 0 0

CMD: shadow-7vm9.shadow.whamcloud.com grep -c /mnt/lustre' ' /proc/mounts

Stopping client shadow-7vm9.shadow.whamcloud.com /mnt/lustre (opts:)

CMD: shadow-7vm9.shadow.whamcloud.com lsof -t /mnt/lustre

CMD: shadow-7vm9.shadow.whamcloud.com umount /mnt/lustre 2>&1

CMD: shadow-7vm8 /usr/sbin/lctl set\_param fail\_val=1 fail\_loc=0x1601

fail\_val=1

fail\_loc=0x1601

CMD: shadow-7vm8 /usr/sbin/lctl lfsck\_start -M lustre-MDT0000 -t namespace -r

Started LFSCK on the device lustre-MDT0000: scrub namespace

CMD: shadow-7vm8 /usr/sbin/lctl get\_param -n mdd.lustre-MDT0000.lfsck\_namespace

stop mds1

CMD: shadow-7vm8 grep -c /mnt/mds1' ' /proc/mounts

CMD: shadow-7vm8 umount -d /mnt/mds1

CMD: shadow-7vm8 lsmod | grep lnet > /dev/null && lctl dl | grep ' ST '

start mds1

CMD: shadow-7vm8 mkdir -p /mnt/mds1

CMD: shadow-7vm8 test -b /dev/lvm-Role\_MDS/P1

CMD: shadow-7vm8 mkdir -p /mnt/mds1; mount -t lustre -o user\_xattr  
/dev/lvm-Role\_MDS/P1 /mnt/mds1

CMD: shadow-7vm8

PATH=/usr/lib64/lustre/tests:/usr/lib/lustre/tests:/usr/lib64/lustre/tests:/opt/iozone/bin:/  
opt/iozone/bin:/usr/lib64/lustre/tests/mpi:/usr/lib64/lustre/tests/racer:/usr/lib64/lustre/.  
./lustre-iokit/sgpdd-  
survey:/usr/lib64/lustre/tests:/usr/lib64/lustre/utills/gss:/usr/lib64/lustre/utills:/usr/lib6  
4/openmpi/bin:/usr/bin:/bin:/usr/sbin:/sbin::/sbin:/bin:/usr/sbin: NAME=autotest\_config sh  
rpc.sh set\_default\_debug \"vfstrace rpctrace dltrace neterror ha config ioctl super lfsck\"  
\"all -lnet -lnd -pinger\" 4

CMD: shadow-7vm8 e2label /dev/lvm-Role\_MDS/P1 2>/dev/null

CMD: shadow-7vm8 /usr/sbin/lctl set\_param fail\_loc=0 fail\_val=0

fail\_loc=0

fail\_val=0

CMD: shadow-7vm8 /usr/sbin/lctl get\_param -n mdd.lustre-MDT0000.lfsck\_namespace |  
awk '/^status/ { print \\$2 }'

CMD: shadow-7vm8 /usr/sbin/lctl get\_param -n mdd.lustre-MDT0000.lfsck\_namespace |  
awk '/^status/ { print \\$2 }'

Waiting 30 secs for update

CMD: shadow-7vm8 /usr/sbin/lctl get\_param -n mdd.lustre-MDT0000.lfsck\_namespace |  
awk '/^status/ { print \\$2 }'

```
Resetting fail_loc on all nodes...CMD: shadow-7vm4,shadow-7vm6,shadow-7vm7,shadow-7vm8,shadow-7vm9.shadow.whamcloud.com lctl set_param -n fail_loc=0 fail_val=0
2>/dev/null || true
```

done.

```
CMD: shadow-7vm4,shadow-7vm6,shadow-7vm7,shadow-7vm8 rc=\${[ -f /proc/sys/lnet/catastrophe ]
&&
```

```
    echo \${< /proc/sys/lnet/catastrophe} || echo 0);
    if [ \${rc} -ne 0 ]; then echo \$(hostname): \${rc}; fi
    exit \${rc}
```

PASS 7a (19s)

```
== sanity-lfsck test 7b: non-stopped LFSCK should auto restarts after MDS remount (2) ==
06:17:52 (1409984272)
```

Checking servers environments

```
CMD: shadow-7vm7 running=\$(grep -c /mnt/ost1' ' /proc/mounts);
```

```
mpts=\$(mount | grep -c /mnt/ost1' ');
```

```
if [ \${running} -ne \${mpts} ]; then
```

```
    echo \$(hostname) env are INSANE!;
```

```
    exit 1;
```

```
fi
```

```
CMD: shadow-7vm7 running=\$(grep -c /mnt/ost2' ' /proc/mounts);
```

```
mpts=\$(mount | grep -c /mnt/ost2' ');
```

```
if [ \${running} -ne \${mpts} ]; then
```

```
    echo \$(hostname) env are INSANE!;
```

```
    exit 1;
```

```
fi
```

```
CMD: shadow-7vm7 running=\$(grep -c /mnt/ost3' ' /proc/mounts);
```

```
mpts=\$(mount | grep -c /mnt/ost3' ');
```

```
if [ \${running} -ne \${mpts} ]; then
```

```
    echo \$(hostname) env are INSANE!;
```

```
    exit 1;
```

```
fi
```

```
CMD: shadow-7vm7 running=\$(grep -c /mnt/ost4' ' /proc/mounts);
```

```
mpts=\$(mount | grep -c /mnt/ost4' ');
```

```
if [ \${running} -ne \${mpts} ]; then
```

```
    echo \$(hostname) env are INSANE!;
```

```
    exit 1;
```

```
fi
```

```
CMD: shadow-7vm8 running=\$(grep -c /mnt/mds1' ' /proc/mounts);
```

```
mpts=\$(mount | grep -c /mnt/mds1' ');
```

```
if [ \${running} -ne \${mpts} ]; then
```

```
    echo \$(hostname) env are INSANE!;
```

```
    exit 1;
```

```
fi
```

```
CMD: shadow-7vm4 running=\$(grep -c /mnt/mds2' ' /proc/mounts);
```

```
mpts=\$(mount | grep -c /mnt/mds2' ');
```

```
if [ \${running} -ne \${mpts} ]; then
```

```
    echo \$(hostname) env are INSANE!;
```

```
    exit 1;
```

```

fi
CMD: shadow-7vm4 running=\$(grep -c /mnt/mds3' ' /proc/mounts);
mpts=\$(mount | grep -c /mnt/mds3' ');
if [ \$running -ne \$mpts ]; then
    echo \$(hostname) env are INSANE!;
    exit 1;
fi
CMD: shadow-7vm4 running=\$(grep -c /mnt/mds4' ' /proc/mounts);
mpts=\$(mount | grep -c /mnt/mds4' ');
if [ \$running -ne \$mpts ]; then
    echo \$(hostname) env are INSANE!;
    exit 1;
fi
CMD: shadow-7vm8 running=\$(grep -c /mnt/mds1' ' /proc/mounts);
mpts=\$(mount | grep -c /mnt/mds1' ');
if [ \$running -ne \$mpts ]; then
    echo \$(hostname) env are INSANE!;
    exit 1;
fi
Checking clients shadow-7vm6,shadow-7vm9.shadow.whamcloud.com environments
CMD: shadow-7vm6,shadow-7vm9.shadow.whamcloud.com running=\$(grep -c /mnt/lustre' '
/proc/mounts);
mpts=\$(mount | grep -c /mnt/lustre' ');
if [ \$running -ne \$mpts ]; then
    echo \$(hostname) env are INSANE!;
    exit 1;
fi
CMD: shadow-7vm6,shadow-7vm9.shadow.whamcloud.com running=\$(grep -c /mnt/lustre2' '
/proc/mounts);
mpts=\$(mount | grep -c /mnt/lustre2' ');
if [ \$running -ne \$mpts ]; then
    echo \$(hostname) env are INSANE!;
    exit 1;
fi
Loading modules from /usr/lib64/lustre
detected 2 online CPUs by sysfs
Force libcfcs to create 2 CPU partitions
debug=vfstrace rpctrace dlmtrace neterror ha config          ioctl super lfsck
subsystem_debug=all -lnet -lnd -pinger
Setup mgs, mdt, osts
CMD: shadow-7vm8 mkdir -p /mnt/mds1
CMD: shadow-7vm8 test -b /dev/lvm-Role_MDS/P1
Starting mds1: /dev/lvm-Role_MDS/P1 /mnt/mds1
CMD: shadow-7vm8 mkdir -p /mnt/mds1; mount -t lustre
/dev/lvm-Role_MDS/P1 /mnt/mds1
shadow-7vm8: mount.lustre: according to /etc/mtab /dev/mapper/lvm--Role_MDS-P1 is already
mounted on /mnt/mds1
Start of /dev/lvm-Role_MDS/P1 on mds1 failed 17
CMD: shadow-7vm4 mkdir -p /mnt/mds2

```

```
CMD: shadow-7vm4 test -b /dev/lvm-Role_MDS/P2
Starting mds2: /dev/lvm-Role_MDS/P2 /mnt/mds2
CMD: shadow-7vm4 mkdir -p /mnt/mds2; mount -t lustre
/dev/lvm-Role_MDS/P2 /mnt/mds2
shadow-7vm4: mount.lustre: according to /etc/mtab /dev/mapper/lvm--Role_MDS-P2 is already
mounted on /mnt/mds2
Start of /dev/lvm-Role_MDS/P2 on mds2 failed 17
CMD: shadow-7vm4 mkdir -p /mnt/mds3
CMD: shadow-7vm4 test -b /dev/lvm-Role_MDS/P3
Starting mds3: /dev/lvm-Role_MDS/P3 /mnt/mds3
CMD: shadow-7vm4 mkdir -p /mnt/mds3; mount -t lustre
/dev/lvm-Role_MDS/P3 /mnt/mds3
shadow-7vm4: mount.lustre: according to /etc/mtab /dev/mapper/lvm--Role_MDS-P3 is already
mounted on /mnt/mds3
Start of /dev/lvm-Role_MDS/P3 on mds3 failed 17
CMD: shadow-7vm4 mkdir -p /mnt/mds4
CMD: shadow-7vm4 test -b /dev/lvm-Role_MDS/P4
Starting mds4: /dev/lvm-Role_MDS/P4 /mnt/mds4
CMD: shadow-7vm4 mkdir -p /mnt/mds4; mount -t lustre
/dev/lvm-Role_MDS/P4 /mnt/mds4
shadow-7vm4: mount.lustre: according to /etc/mtab /dev/mapper/lvm--Role_MDS-P4 is already
mounted on /mnt/mds4
Start of /dev/lvm-Role_MDS/P4 on mds4 failed 17
CMD: shadow-7vm7 mkdir -p /mnt/ost1
CMD: shadow-7vm7 test -b /dev/lvm-Role_OSS/P1
Starting ost1: /dev/lvm-Role_OSS/P1 /mnt/ost1
CMD: shadow-7vm7 mkdir -p /mnt/ost1; mount -t lustre
/dev/lvm-Role_OSS/P1 /mnt/ost1
shadow-7vm7: mount.lustre: according to /etc/mtab /dev/mapper/lvm--Role_OSS-P1 is already
mounted on /mnt/ost1
Start of /dev/lvm-Role_OSS/P1 on ost1 failed 17
CMD: shadow-7vm7 mkdir -p /mnt/ost2
CMD: shadow-7vm7 test -b /dev/lvm-Role_OSS/P2
Starting ost2: /dev/lvm-Role_OSS/P2 /mnt/ost2
CMD: shadow-7vm7 mkdir -p /mnt/ost2; mount -t lustre
/dev/lvm-Role_OSS/P2 /mnt/ost2
shadow-7vm7: mount.lustre: according to /etc/mtab /dev/mapper/lvm--Role_OSS-P2 is already
mounted on /mnt/ost2
Start of /dev/lvm-Role_OSS/P2 on ost2 failed 17
CMD: shadow-7vm7 mkdir -p /mnt/ost3
CMD: shadow-7vm7 test -b /dev/lvm-Role_OSS/P3
Starting ost3: /dev/lvm-Role_OSS/P3 /mnt/ost3
CMD: shadow-7vm7 mkdir -p /mnt/ost3; mount -t lustre
/dev/lvm-Role_OSS/P3 /mnt/ost3
shadow-7vm7: mount.lustre: according to /etc/mtab /dev/mapper/lvm--Role_OSS-P3 is already
mounted on /mnt/ost3
Start of /dev/lvm-Role_OSS/P3 on ost3 failed 17
CMD: shadow-7vm7 mkdir -p /mnt/ost4
CMD: shadow-7vm7 test -b /dev/lvm-Role_OSS/P4
Starting ost4: /dev/lvm-Role_OSS/P4 /mnt/ost4
```

```

CMD: shadow-7vm7 mkdir -p /mnt/ost4; mount -t lustre
/dev/lvm-Role_OSS/P4 /mnt/ost4
shadow-7vm7: mount.lustre: according to /etc/mstab /dev/mapper/lvm--Role_OSS-P4 is already
mounted on /mnt/ost4
Start of /dev/lvm-Role_OSS/P4 on ost4 failed 17
Starting client: shadow-7vm9.shadow.whamcloud.com: -o user_xattr,flock shadow-
7vm8@tcp:/lustre /mnt/lustre
CMD: shadow-7vm9.shadow.whamcloud.com mkdir -p /mnt/lustre
CMD: shadow-7vm9.shadow.whamcloud.com mount -t lustre -o user_xattr,flock shadow-
7vm8@tcp:/lustre /mnt/lustre
Starting client shadow-7vm6,shadow-7vm9.shadow.whamcloud.com: -o user_xattr,flock shadow-
7vm8@tcp:/lustre /mnt/lustre
CMD: shadow-7vm6,shadow-7vm9.shadow.whamcloud.com
running=\$(mount | grep -c /mnt/lustre' ');
rc=0;
if [ \$running -eq 0 ] ; then
    mkdir -p /mnt/lustre;
    mount -t lustre -o user_xattr,flock shadow-7vm8@tcp:/lustre /mnt/lustre;
    rc=\$?;
fi;
exit \$rc
Started clients shadow-7vm6,shadow-7vm9.shadow.whamcloud.com:
CMD: shadow-7vm6,shadow-7vm9.shadow.whamcloud.com mount | grep /mnt/lustre' '
shadow-7vm8@tcp:/lustre on /mnt/lustre type lustre (rw,user_xattr,flock)
shadow-7vm8@tcp:/lustre on /mnt/lustre type lustre (rw,user_xattr,flock)
CMD: shadow-7vm6
PATH=/usr/lib64/lustre/tests:/usr/lib/lustre/tests:/usr/lib64/lustre/tests:/opt/iozone/bin:/
opt/iozone/bin:/usr/lib64/lustre/tests/mpi:/usr/lib64/lustre/tests/racer:/usr/lib64/lustre/.
./lustre-iokit/sgpdd-
survey:/usr/lib64/lustre/tests:/usr/lib64/lustre/utills/gss:/usr/lib64/lustre/utills:/usr/lib6
4/openmpi/bin:/usr/bin:/bin:/usr/sbin:/sbin:/:/sbin:/bin:/usr/sbin: NAME=autotest_config sh
rpc.sh set_default_debug "\vfstrace rpctrace dltrace neterror ha config ioctl super lfsck\"
\"all -lnet -lnd -pinger\" 4
CMD: shadow-7vm8 lctl get_param -n timeout
Using TIMEOUT=20
CMD: shadow-7vm8 lctl dl | grep ' IN osc ' 2>/dev/null | wc -l
CMD: shadow-7vm9.shadow.whamcloud.com lctl dl | grep ' IN osc ' 2>/dev/null | wc -l
disable quota as required
preparing... 2 * 2 files will be created Sat Sep 6 06:18:11 UTC 2014.
total: 2 creates in 0.00 seconds: 621.70 creates/second
total: 2 creates in 0.00 seconds: 685.62 creates/second
total: 2 creates in 0.00 seconds: 617.45 creates/second
prepared Sat Sep 6 06:18:12 UTC 2014.
CMD: shadow-7vm8 /usr/sbin/lctl set_param fail_loc=0x1604
fail_loc=0x1604
CMD: shadow-7vm8 /usr/sbin/lctl set_param fail_val=1 fail_loc=0x1602
fail_val=1
fail_loc=0x1602
CMD: shadow-7vm8 /usr/sbin/lctl lfsck_start -M lustre-MDT0000 -t namespace -r
Started LFSCK on the device lustre-MDT0000: scrub namespace
CMD: shadow-7vm8 /usr/sbin/lctl get_param -n mdd.lustre-MDT0000.lfsck_namespace |

```

```

        awk '/^status/ { print \$2 }'
CMD: shadow-7vm8 /usr/sbin/lctl get_param -n          mdd.lustre-MDT0000.lfsck_namespace |
        awk '/^status/ { print \$2 }'
stop mds1
CMD: shadow-7vm8 grep -c /mnt/mds1' ' /proc/mounts
CMD: shadow-7vm8 umount -d /mnt/mds1
CMD: shadow-7vm8 lsmod | grep lnet > /dev/null && lctl dl | grep ' ST '
start mds1
CMD: shadow-7vm8 mkdir -p /mnt/mds1
CMD: shadow-7vm8 test -b /dev/lvm-Role_MDS/P1
CMD: shadow-7vm8 mkdir -p /mnt/mds1; mount -t lustre -o user_xattr
/dev/lvm-Role_MDS/P1 /mnt/mds1
CMD: shadow-7vm8
PATH=/usr/lib64/lustre/tests:/usr/lib/lustre/tests:/usr/lib64/lustre/tests:/opt/iozone/bin:/
opt/iozone/bin:/usr/lib64/lustre/tests/mpi:/usr/lib64/lustre/tests/racer:/usr/lib64/lustre/.
./lustre-iokit/sgpdd-
survey:/usr/lib64/lustre/tests:/usr/lib64/lustre/utills/gss:/usr/lib64/lustre/utills:/usr/lib6
4/openmpi/bin:/usr/bin:/bin:/usr/sbin:/sbin::/sbin:/bin:/usr/sbin: NAME=autotest_config sh
rpc.sh set_default_debug \"vfstrace rpctrace dlmtrace neterror ha config ioctl super lfsck\"
\"all -lnet -lnd -pingr\" 4
CMD: shadow-7vm8 e2label /dev/lvm-Role_MDS/P1 2>/dev/null
CMD: shadow-7vm8 /usr/sbin/lctl set_param fail_loc=0 fail_val=0
fail_loc=0
fail_val=0
CMD: shadow-7vm8 /usr/sbin/lctl get_param -n          mdd.lustre-MDT0000.lfsck_namespace |
        awk '/^status/ { print \$2 }'
CMD: shadow-7vm8 /usr/sbin/lctl get_param -n          mdd.lustre-MDT0000.lfsck_namespace |
        awk '/^status/ { print \$2 }'
Waiting 30 secs for update
CMD: shadow-7vm8 /usr/sbin/lctl get_param -n          mdd.lustre-MDT0000.lfsck_namespace |
        awk '/^status/ { print \$2 }'
CMD: shadow-7vm8 /usr/sbin/lctl get_param -n          mdd.lustre-MDT0000.lfsck_namespace |
        awk '/^status/ { print \$2 }'
CMD: shadow-7vm8 /usr/sbin/lctl get_param -n          mdd.lustre-MDT0000.lfsck_namespace |
        awk '/^status/ { print \$2 }'
CMD: shadow-7vm8 /usr/sbin/lctl get_param -n          mdd.lustre-MDT0000.lfsck_namespace |
        awk '/^status/ { print \$2 }'
Updated after 4s: wanted 'completed' got 'completed'
Resetting fail_loc on all nodes...CMD: shadow-7vm4,shadow-7vm6,shadow-7vm7,shadow-
7vm8,shadow-7vm9.shadow.whamcloud.com lctl set_param -n fail_loc=0          fail_val=0
2>/dev/null || true
done.
CMD: shadow-7vm4,shadow-7vm6,shadow-7vm7,shadow-7vm8 rc=\$([ -f /proc/sys/lnet/catastrophe ]
&&
        echo \$(< /proc/sys/lnet/catastrophe) || echo 0);
        if [ \$rc -ne 0 ]; then echo \$(hostname): \$rc; fi
        exit \$rc
PASS 7b (37s)

== sanity-lfsck test 9a: LFSCK speed control (1) == 06:21:11 (1409984471)
Checking servers environments

```

```
CMD: shadow-7vm7 running=\$(grep -c /mnt/ost1' ' /proc/mounts);
mpts=\$(mount | grep -c /mnt/ost1' ');
if [ \$running -ne \$mpts ]; then
    echo \$(hostname) env are INSANE!;
    exit 1;
fi
CMD: shadow-7vm7 running=\$(grep -c /mnt/ost2' ' /proc/mounts);
mpts=\$(mount | grep -c /mnt/ost2' ');
if [ \$running -ne \$mpts ]; then
    echo \$(hostname) env are INSANE!;
    exit 1;
fi
CMD: shadow-7vm7 running=\$(grep -c /mnt/ost3' ' /proc/mounts);
mpts=\$(mount | grep -c /mnt/ost3' ');
if [ \$running -ne \$mpts ]; then
    echo \$(hostname) env are INSANE!;
    exit 1;
fi
CMD: shadow-7vm7 running=\$(grep -c /mnt/ost4' ' /proc/mounts);
mpts=\$(mount | grep -c /mnt/ost4' ');
if [ \$running -ne \$mpts ]; then
    echo \$(hostname) env are INSANE!;
    exit 1;
fi
CMD: shadow-7vm8 running=\$(grep -c /mnt/mds1' ' /proc/mounts);
mpts=\$(mount | grep -c /mnt/mds1' ');
if [ \$running -ne \$mpts ]; then
    echo \$(hostname) env are INSANE!;
    exit 1;
fi
CMD: shadow-7vm4 running=\$(grep -c /mnt/mds2' ' /proc/mounts);
mpts=\$(mount | grep -c /mnt/mds2' ');
if [ \$running -ne \$mpts ]; then
    echo \$(hostname) env are INSANE!;
    exit 1;
fi
CMD: shadow-7vm4 running=\$(grep -c /mnt/mds3' ' /proc/mounts);
mpts=\$(mount | grep -c /mnt/mds3' ');
if [ \$running -ne \$mpts ]; then
    echo \$(hostname) env are INSANE!;
    exit 1;
fi
CMD: shadow-7vm4 running=\$(grep -c /mnt/mds4' ' /proc/mounts);
mpts=\$(mount | grep -c /mnt/mds4' ');
if [ \$running -ne \$mpts ]; then
    echo \$(hostname) env are INSANE!;
    exit 1;
fi
```

```

CMD: shadow-7vm8 running=\$(grep -c /mnt/mds1' ' /proc/mounts);
mpts=\$(mount | grep -c /mnt/mds1' ');
if [ \$running -ne \$mpts ]; then
    echo \$(hostname) env are INSANE!;
    exit 1;
fi
Checking clients shadow-7vm6,shadow-7vm9.shadow.whamcloud.com environments
CMD: shadow-7vm6,shadow-7vm9.shadow.whamcloud.com running=\$(grep -c /mnt/lustre' '
/proc/mounts);
mpts=\$(mount | grep -c /mnt/lustre' ');
if [ \$running -ne \$mpts ]; then
    echo \$(hostname) env are INSANE!;
    exit 1;
fi
CMD: shadow-7vm6,shadow-7vm9.shadow.whamcloud.com running=\$(grep -c /mnt/lustre2' '
/proc/mounts);
mpts=\$(mount | grep -c /mnt/lustre2' ');
if [ \$running -ne \$mpts ]; then
    echo \$(hostname) env are INSANE!;
    exit 1;
fi
Loading modules from /usr/lib64/lustre
detected 2 online CPUs by sysfs
Force libcfs to create 2 CPU partitions
debug=vfstrace rpctrace dlmtrace neterror ha config          ioctl super lfsck
subsystem_debug=all -lnet -lnd -pinger
Setup mgs, mdt, osts
CMD: shadow-7vm8 mkdir -p /mnt/mds1
CMD: shadow-7vm8 test -b /dev/lvm-Role_MDS/P1
Starting mds1: /dev/lvm-Role_MDS/P1 /mnt/mds1
CMD: shadow-7vm8 mkdir -p /mnt/mds1; mount -t lustre
/dev/lvm-Role_MDS/P1 /mnt/mds1
shadow-7vm8: mount.lustre: according to /etc/mtab /dev/mapper/lvm--Role_MDS-P1 is already
mounted on /mnt/mds1
Start of /dev/lvm-Role_MDS/P1 on mds1 failed 17
CMD: shadow-7vm4 mkdir -p /mnt/mds2
CMD: shadow-7vm4 test -b /dev/lvm-Role_MDS/P2
Starting mds2: /dev/lvm-Role_MDS/P2 /mnt/mds2
CMD: shadow-7vm4 mkdir -p /mnt/mds2; mount -t lustre
/dev/lvm-Role_MDS/P2 /mnt/mds2
shadow-7vm4: mount.lustre: according to /etc/mtab /dev/mapper/lvm--Role_MDS-P2 is already
mounted on /mnt/mds2
Start of /dev/lvm-Role_MDS/P2 on mds2 failed 17
CMD: shadow-7vm4 mkdir -p /mnt/mds3
CMD: shadow-7vm4 test -b /dev/lvm-Role_MDS/P3
Starting mds3: /dev/lvm-Role_MDS/P3 /mnt/mds3
CMD: shadow-7vm4 mkdir -p /mnt/mds3; mount -t lustre
/dev/lvm-Role_MDS/P3 /mnt/mds3
shadow-7vm4: mount.lustre: according to /etc/mtab /dev/mapper/lvm--Role_MDS-P3 is already
mounted on /mnt/mds3

```



```
Start of /dev/lvm-Role_MDS/P3 on mds3 failed 17
CMD: shadow-7vm4 mkdir -p /mnt/mds4
CMD: shadow-7vm4 test -b /dev/lvm-Role_MDS/P4
Starting mds4: /dev/lvm-Role_MDS/P4 /mnt/mds4
CMD: shadow-7vm4 mkdir -p /mnt/mds4; mount -t lustre
/dev/lvm-Role_MDS/P4 /mnt/mds4
shadow-7vm4: mount.lustre: according to /etc/mtab /dev/mapper/lvm--Role_MDS-P4 is already
mounted on /mnt/mds4
Start of /dev/lvm-Role_MDS/P4 on mds4 failed 17
CMD: shadow-7vm7 mkdir -p /mnt/ost1
CMD: shadow-7vm7 test -b /dev/lvm-Role_OSS/P1
Starting ost1: /dev/lvm-Role_OSS/P1 /mnt/ost1
CMD: shadow-7vm7 mkdir -p /mnt/ost1; mount -t lustre
/dev/lvm-Role_OSS/P1 /mnt/ost1
shadow-7vm7: mount.lustre: according to /etc/mtab /dev/mapper/lvm--Role_OSS-P1 is already
mounted on /mnt/ost1
Start of /dev/lvm-Role_OSS/P1 on ost1 failed 17
CMD: shadow-7vm7 mkdir -p /mnt/ost2
CMD: shadow-7vm7 test -b /dev/lvm-Role_OSS/P2
Starting ost2: /dev/lvm-Role_OSS/P2 /mnt/ost2
CMD: shadow-7vm7 mkdir -p /mnt/ost2; mount -t lustre
/dev/lvm-Role_OSS/P2 /mnt/ost2
shadow-7vm7: mount.lustre: according to /etc/mtab /dev/mapper/lvm--Role_OSS-P2 is already
mounted on /mnt/ost2
Start of /dev/lvm-Role_OSS/P2 on ost2 failed 17
CMD: shadow-7vm7 mkdir -p /mnt/ost3
CMD: shadow-7vm7 test -b /dev/lvm-Role_OSS/P3
Starting ost3: /dev/lvm-Role_OSS/P3 /mnt/ost3
CMD: shadow-7vm7 mkdir -p /mnt/ost3; mount -t lustre
/dev/lvm-Role_OSS/P3 /mnt/ost3
shadow-7vm7: mount.lustre: according to /etc/mtab /dev/mapper/lvm--Role_OSS-P3 is already
mounted on /mnt/ost3
Start of /dev/lvm-Role_OSS/P3 on ost3 failed 17
CMD: shadow-7vm7 mkdir -p /mnt/ost4
CMD: shadow-7vm7 test -b /dev/lvm-Role_OSS/P4
Starting ost4: /dev/lvm-Role_OSS/P4 /mnt/ost4
CMD: shadow-7vm7 mkdir -p /mnt/ost4; mount -t lustre
/dev/lvm-Role_OSS/P4 /mnt/ost4
shadow-7vm7: mount.lustre: according to /etc/mtab /dev/mapper/lvm--Role_OSS-P4 is already
mounted on /mnt/ost4
Start of /dev/lvm-Role_OSS/P4 on ost4 failed 17
Starting client: shadow-7vm9.shadow.whamcloud.com: -o user_xattr,flock shadow-
7vm8@tcp:/lustre /mnt/lustre
CMD: shadow-7vm9.shadow.whamcloud.com mkdir -p /mnt/lustre
CMD: shadow-7vm9.shadow.whamcloud.com mount -t lustre -o user_xattr,flock shadow-
7vm8@tcp:/lustre /mnt/lustre
Starting client shadow-7vm6,shadow-7vm9.shadow.whamcloud.com: -o user_xattr,flock shadow-
7vm8@tcp:/lustre /mnt/lustre
CMD: shadow-7vm6,shadow-7vm9.shadow.whamcloud.com
running=\$(mount | grep -c /mnt/lustre' ');
rc=0;
```

```

if [ \${running -eq 0} ] ; then
    mkdir -p /mnt/lustre;
    mount -t lustre -o user_xattr,flock shadow-7vm8@tcp:/lustre /mnt/lustre;
    rc=\${?};
fi;
exit \${rc}
Started clients shadow-7vm6,shadow-7vm9.shadow.whamcloud.com:
CMD: shadow-7vm6,shadow-7vm9.shadow.whamcloud.com mount | grep /mnt/lustre' '
shadow-7vm8@tcp:/lustre on /mnt/lustre type lustre (rw,user_xattr,flock)
shadow-7vm8@tcp:/lustre on /mnt/lustre type lustre (rw,user_xattr,flock)
CMD: shadow-7vm6
PATH=/usr/lib64/lustre/tests:/usr/lib/lustre/tests:/usr/lib64/lustre/tests:/opt/iozone/bin:/
opt/iozone/bin:/usr/lib64/lustre/tests/mpi:/usr/lib64/lustre/tests/racer:/usr/lib64/lustre/.
./lustre-iokit/sgpdd-
survey:/usr/lib64/lustre/tests:/usr/lib64/lustre/utills/gss:/usr/lib64/lustre/utills:/usr/lib6
4/openmpi/bin:/usr/bin:/bin:/usr/sbin:/sbin::/sbin:/bin:/usr/sbin: NAME=autotest_config sh
rpc.sh set_default_debug "\vfstrace rpctrace dlmtrace neterror ha config ioctl super lfsck\"
\"all -lnet -lnd -pinger\" 4
CMD: shadow-7vm8 lctl get_param -n timeout
Using TIMEOUT=20
CMD: shadow-7vm8 lctl dl | grep ' IN osc ' 2>/dev/null | wc -l
CMD: shadow-7vm9.shadow.whamcloud.com lctl dl | grep ' IN osc ' 2>/dev/null | wc -l
disable quota as required
preparing... 70 * 70 files will be created Sat Sep 6 06:21:32 UTC 2014.
total: 70 creates in 0.11 seconds: 656.77 creates/second
total: 70 creates in 0.10 seconds: 692.18 creates/second
total: 70 creates in 0.16 seconds: 428.09 creates/second
prepared Sat Sep 6 06:21:41 UTC 2014.
CMD: shadow-7vm8 /usr/sbin/lctl lfsck_start -M lustre-MDT0000 -t namespace -r -s 100
Started LFSCK on the device lustre-MDT0000: scrub namespace
CMD: shadow-7vm8 /usr/sbin/lctl get_param -n mdd.lustre-MDT0000.lfsck_namespace
CMD: shadow-7vm8 /usr/sbin/lctl get_param -n mdd.lustre-MDT0000.lfsck_namespace
CMD: shadow-7vm8 /usr/sbin/lctl set_param -n mdd.lustre-MDT0000.lfsck_speed_limit 300
CMD: shadow-7vm8 /usr/sbin/lctl get_param -n mdd.lustre-MDT0000.lfsck_namespace
CMD: shadow-7vm8 /usr/sbin/lctl set_param -n mdd.lustre-MDT0000.lfsck_speed_limit 0
CMD: shadow-7vm8 /usr/sbin/lctl get_param -n mdd.lustre-MDT0000.lfsck_namespace|
awk '/^status/ { print \$2 }'
CMD: shadow-7vm8 /usr/sbin/lctl get_param -n mdd.lustre-MDT0000.lfsck_namespace|
awk '/^status/ { print \$2 }'
Resetting fail_loc on all nodes...CMD: shadow-7vm4,shadow-7vm6,shadow-7vm7,shadow-
7vm8,shadow-7vm9.shadow.whamcloud.com lctl set_param -n fail_loc=0 fail_val=0
2>/dev/null || true
done.
CMD: shadow-7vm4,shadow-7vm6,shadow-7vm7,shadow-7vm8 rc=\${([ -f /proc/sys/lnet/catastrophe ]
&&
    echo \${(< /proc/sys/lnet/catastrophe) || echo 0};
    if [ \${rc} -ne 0 ]; then echo \$(hostname): \${rc}; fi
    exit \${rc}
PASS 9a (53s)

```

```

== sanity-lfsck test 9b: LFSCK speed control (2) == 06:22:04 (1409984524)

```

```

preparing... 0 * 0 files will be created Sat Sep 6 06:22:16 UTC 2014.
prepared Sat Sep 6 06:22:17 UTC 2014.
Preparing another 50 * 50 files (with error) at Sat Sep 6 06:22:17 UTC 2014.
CMD: shadow-7vm8 /usr/sbin/lctl set_param fail_loc=0x1604
fail_loc=0x1604
total: 50 creates in 0.08 seconds: 597.49 creates/second
total: 50 creates in 0.07 seconds: 719.18 creates/second
CMD: shadow-7vm8 /usr/sbin/lctl set_param fail_loc=0x160c
fail_loc=0x160c
CMD: shadow-7vm8 /usr/sbin/lctl lfsck_start -M lustre-MDT0000 -t namespace -r
Started LFSCK on the device lustre-MDT0000: scrub namespace
CMD: shadow-7vm8 /usr/sbin/lctl get_param -n mdd.lustre-MDT0000.lfsck_namespace |
awk '/^status/ { print \$2 }'
CMD: shadow-7vm8 /usr/sbin/lctl get_param -n mdd.lustre-MDT0000.lfsck_namespace |
awk '/^status/ { print \$2 }'
CMD: shadow-7vm8 /usr/sbin/lctl set_param fail_loc=0
fail_loc=0
Prepared at Sat Sep 6 06:22:23 UTC 2014.
CMD: shadow-7vm8 /usr/sbin/lctl lfsck_start -M lustre-MDT0000 -t namespace -s 50
Started LFSCK on the device lustre-MDT0000: scrub namespace
CMD: shadow-7vm8 /usr/sbin/lctl get_param -n mdd.lustre-MDT0000.lfsck_namespace
CMD: shadow-7vm8 /usr/sbin/lctl get_param -n mdd.lustre-MDT0000.lfsck_namespace
CMD: shadow-7vm8 /usr/sbin/lctl set_param -n mdd.lustre-MDT0000.lfsck_speed_limit 150
CMD: shadow-7vm8 /usr/sbin/lctl get_param -n mdd.lustre-MDT0000.lfsck_namespace
CMD: shadow-7vm8 /usr/sbin/lctl set_param -n mdd.lustre-MDT0000.lfsck_speed_limit 0
CMD: shadow-7vm8 /usr/sbin/lctl get_param -n mdd.lustre-MDT0000.lfsck_namespace |
awk '/^status/ { print \$2 }'
CMD: shadow-7vm8 /usr/sbin/lctl get_param -n mdd.lustre-MDT0000.lfsck_namespace |
awk '/^status/ { print \$2 }'
Resetting fail_loc on all nodes...CMD: shadow-7vm4,shadow-7vm6,shadow-7vm7,shadow-
7vm8,shadow-7vm9.shadow.whamcloud.com lctl set_param -n fail_loc=0 fail_val=0
2>/dev/null || true
done.
CMD: shadow-7vm4,shadow-7vm6,shadow-7vm7,shadow-7vm8 rc=\$([ -f /proc/sys/lnet/catastrophe ]
&&
echo \$(< /proc/sys/lnet/catastrophe) || echo 0);
if [ \$rc -ne 0 ]; then echo \$(hostname): \$rc; fi
exit \$rc
PASS 9b (41s)

== sanity-lfsck test 10: System is available during LFSCK scanning == 06:22:46 (1409984566)
preparing... 1 * 1 files will be created Sat Sep 6 06:22:52 UTC 2014.
total: 1 creates in 0.00 seconds: 622.30 creates/second
total: 1 creates in 0.00 seconds: 594.52 creates/second
total: 1 creates in 0.00 seconds: 593.42 creates/second
prepared Sat Sep 6 06:22:53 UTC 2014.
Preparing more files with error at Sat Sep 6 06:22:53 UTC 2014.
CMD: shadow-7vm8 /usr/sbin/lctl set_param fail_loc=0x1603
fail_loc=0x1603

```

```
CMD: shadow-7vm8 /usr/sbin/lctl set_param fail_loc=0x1604
fail_loc=0x1604
CMD: shadow-7vm8 /usr/sbin/lctl set_param fail_loc=0
fail_loc=0
Prepared at Sat Sep 6 06:23:23 UTC 2014.
10.1.4.75@tcp:/lustre /mnt/lustre lustre rw,flock,user_xattr 0 0
CMD: shadow-7vm9.shadow.whamcloud.com grep -c /mnt/lustre' ' /proc/mounts
Stopping client shadow-7vm9.shadow.whamcloud.com /mnt/lustre (opts:)
CMD: shadow-7vm9.shadow.whamcloud.com lsof -t /mnt/lustre
CMD: shadow-7vm9.shadow.whamcloud.com umount /mnt/lustre 2>&1
Starting client: shadow-7vm9.shadow.whamcloud.com: -o user_xattr,flock shadow-
7vm8@tcp:/lustre /mnt/lustre
CMD: shadow-7vm9.shadow.whamcloud.com mkdir -p /mnt/lustre
CMD: shadow-7vm9.shadow.whamcloud.com mount -t lustre -o user_xattr,flock shadow-
7vm8@tcp:/lustre /mnt/lustre
CMD: shadow-7vm8 /usr/sbin/lctl lfsck_start -M lustre-MDT0000 -t namespace -r -s 100
Started LFSCK on the device lustre-MDT0000: scrub namespace
CMD: shadow-7vm8 /usr/sbin/lctl get_param -n mdd.lustre-MDT0000.lfsck_namespace
CMD: shadow-7vm8 /usr/sbin/lctl get_param -n mdd.lustre-MDT0000.lfsck_namespace
CMD: shadow-7vm8 /usr/sbin/lctl set_param -n mdd.lustre-MDT0000.lfsck_speed_limit 0
CMD: shadow-7vm8 /usr/sbin/lctl get_param -n mdd.lustre-MDT0000.lfsck_namespace |
awk '/^status/ { print \$2 }'
CMD: shadow-7vm8 /usr/sbin/lctl get_param -n mdd.lustre-MDT0000.lfsck_namespace |
awk '/^status/ { print \$2 }'
Waiting 32 secs for update
CMD: shadow-7vm8 /usr/sbin/lctl get_param -n mdd.lustre-MDT0000.lfsck_namespace |
awk '/^status/ { print \$2 }'
CMD: shadow-7vm8 /usr/sbin/lctl get_param -n mdd.lustre-MDT0000.lfsck_namespace |
awk '/^status/ { print \$2 }'
CMD: shadow-7vm8 /usr/sbin/lctl get_param -n mdd.lustre-MDT0000.lfsck_namespace |
awk '/^status/ { print \$2 }'
CMD: shadow-7vm8 /usr/sbin/lctl get_param -n mdd.lustre-MDT0000.lfsck_namespace |
awk '/^status/ { print \$2 }'
CMD: shadow-7vm8 /usr/sbin/lctl get_param -n mdd.lustre-MDT0000.lfsck_namespace |
awk '/^status/ { print \$2 }'
Changed after 5s: from 'scanning-phase1' to 'scanning-phase2'
CMD: shadow-7vm8 /usr/sbin/lctl get_param -n mdd.lustre-MDT0000.lfsck_namespace |
awk '/^status/ { print \$2 }'
CMD: shadow-7vm8 /usr/sbin/lctl get_param -n mdd.lustre-MDT0000.lfsck_namespace |
awk '/^status/ { print \$2 }'
CMD: shadow-7vm8 /usr/sbin/lctl get_param -n mdd.lustre-MDT0000.lfsck_namespace |
awk '/^status/ { print \$2 }'
CMD: shadow-7vm8 /usr/sbin/lctl get_param -n mdd.lustre-MDT0000.lfsck_namespace |
awk '/^status/ { print \$2 }'
Updated after 9s: wanted 'completed' got 'completed'
Resetting fail_loc on all nodes...CMD: shadow-7vm4,shadow-7vm6,shadow-7vm7,shadow-
7vm8,shadow-7vm9.shadow.whamcloud.com lctl set_param -n fail_loc=0 fail_val=0
2>/dev/null || true
done.
```

```
CMD: shadow-7vm4,shadow-7vm6,shadow-7vm7,shadow-7vm8 rc=\$( [ -f /proc/sys/lnet/catastrophe ]
&&
    echo \$(< /proc/sys/lnet/catastrophe) || echo 0);
    if [ \${rc} -ne 0 ]; then echo \$(hostname): \${rc}; fi
    exit \${rc}
```

PASS 10 (82s)

== sanity-lfsck test 22a: LFSCK can repair unmatched pairs (1) == 06:33:34 (1409985214)

#####

The parent\_A references the child directory via some name entry,  
but the child directory back references another parent\_B via its  
.. name entry. The parent\_A does not exist. Then the namespace  
LFSCK will repair the child directory's .. name entry.

#####

Inject failure stub on MDT0 to simulate bad dotdot name entry

The dummy's dotdot name entry references the guard.

```
CMD: shadow-7vm8 /usr/sbin/lctl set_param fail_loc=0x161e
```

```
fail_loc=0x161e
```

```
CMD: shadow-7vm8 /usr/sbin/lctl set_param fail_loc=0
```

```
fail_loc=0
```

Trigger namespace LFSCK to repair unmatched pairs

```
CMD: shadow-7vm8 /usr/sbin/lctl lfsck_start -M lustre-MDT0000 -t namespace -A -r
```

Started LFSCK on the device lustre-MDT0000: scrub namespace

```
CMD: shadow-7vm8 /usr/sbin/lctl get_param -n          mdd.lustre-MDT0000.lfsck_namespace |
    awk '/^status/ { print \$2 }'
```

```
CMD: shadow-7vm8 /usr/sbin/lctl get_param -n          mdd.lustre-MDT0000.lfsck_namespace |
    awk '/^status/ { print \$2 }'
```

```
CMD: shadow-7vm8 /usr/sbin/lctl get_param -n mdd.lustre-MDT0000.lfsck_namespace
```

'ls' should success after namespace LFSCK repairing

```
Resetting fail_loc on all nodes...CMD: shadow-7vm4,shadow-7vm6,shadow-7vm7,shadow-
7vm8,shadow-7vm9.shadow.whamcloud.com lctl set_param -n fail_loc=0          fail_val=0
2>/dev/null || true
```

done.

```
CMD: shadow-7vm4,shadow-7vm6,shadow-7vm7,shadow-7vm8 rc=\$( [ -f /proc/sys/lnet/catastrophe ]
&&
```

```
    echo \$(< /proc/sys/lnet/catastrophe) || echo 0);
    if [ \${rc} -ne 0 ]; then echo \$(hostname): \${rc}; fi
    exit \${rc}
```

PASS 22a (3s)

== sanity-lfsck test 22b: LFSCK can repair unmatched pairs (2) == 06:33:37 (1409985217)

#####

The parent\_A references the child directory via the name entry\_B,  
but the child directory back references another parent\_C via its  
.. name entry. The parent\_C exists, but there is no the name  
entry\_B under the parent\_B. Then the namespace LFSCK will repair  
the child directory's .. name entry and its linkEA.

#####

Inject failure stub on MDT0 to simulate bad dotdot name entry

```

and bad linkEA. The dummy's dotdot name entry references the
guard. The dummy's linkEA references a non-existent name entry.
CMD: shadow-7vm8 /usr/sbin/lctl set_param fail_loc=0x161f
fail_loc=0x161f
CMD: shadow-7vm8 /usr/sbin/lctl set_param fail_loc=0
fail_loc=0
fid2ptah should NOT work on the dummy's FID [0x300002b11:0x77:0x0]
Trigger namespace LFSCK to repair unmatched pairs
CMD: shadow-7vm8 /usr/sbin/lctl lfsck_start -M lustre-MDT0000 -t namespace -A -r
Started LFSCK on the device lustre-MDT0000: scrub namespace
CMD: shadow-7vm8 /usr/sbin/lctl get_param -n mdd.lustre-MDT0000.lfsck_namespace |
    awk '/^status/ { print \$2 }'
CMD: shadow-7vm8 /usr/sbin/lctl get_param -n mdd.lustre-MDT0000.lfsck_namespace |
    awk '/^status/ { print \$2 }'
CMD: shadow-7vm8 /usr/sbin/lctl get_param -n mdd.lustre-MDT0000.lfsck_namespace
fid2ptah should work on the dummy's FID [0x300002b11:0x77:0x0] after LFSCK
Resetting fail_loc on all nodes...CMD: shadow-7vm4,shadow-7vm6,shadow-7vm7,shadow-
7vm8,shadow-7vm9.shadow.whamcloud.com lctl set_param -n fail_loc=0 fail_val=0
2>/dev/null || true
done.
CMD: shadow-7vm4,shadow-7vm6,shadow-7vm7,shadow-7vm8 rc=\$([ -f /proc/sys/lnet/catastrophe ]
&&
    echo \$(< /proc/sys/lnet/catastrophe) || echo 0);
    if [ \$rc -ne 0 ]; then echo \$(hostname): \$rc; fi
    exit \$rc

```

PASS 22b (3s)

== sanity-lfsck test 23a: LFSCK can repair dangling name entry (1) == 06:33:41 (1409985221)

#####

The name entry is there, but the MDT-object for such name entry does not exist. The namespace LFSCK should be able to find out and repair the inconsistency as required.

#####

Inject failure stub on MDT1 to simulate dangling name entry

```

CMD: shadow-7vm4 /usr/sbin/lctl set_param fail_loc=0x1620
fail_loc=0x1620

```

```

CMD: shadow-7vm4 /usr/sbin/lctl set_param fail_loc=0
fail_loc=0

```

'ls' should fail because of dangling name entry

Trigger namespace LFSCK to find out dangling name entry

```

CMD: shadow-7vm8 /usr/sbin/lctl lfsck_start -M lustre-MDT0000 -t namespace -A -r
Started LFSCK on the device lustre-MDT0000: scrub namespace

```

```

CMD: shadow-7vm8 /usr/sbin/lctl get_param -n mdd.lustre-MDT0000.lfsck_namespace |
    awk '/^status/ { print \$2 }'

```

```

CMD: shadow-7vm8 /usr/sbin/lctl get_param -n mdd.lustre-MDT0000.lfsck_namespace |
    awk '/^status/ { print \$2 }'

```

```

CMD: shadow-7vm8 /usr/sbin/lctl get_param -n mdd.lustre-MDT0000.lfsck_namespace

```

'ls' should fail because of not create MDT-object by default

Trigger namespace LFSCK again to repair dangling name entry

```

CMD: shadow-7vm8 /usr/sbin/lctl lfsck_start -M lustre-MDT0000 -t namespace -A -r -C
Started LFSCK on the device lustre-MDT0000: scrub namespace
CMD: shadow-7vm8 /usr/sbin/lctl get_param -n          mdd.lustre-MDT0000.lfsck_namespace |
      awk '/^status/ { print \$2 }'
CMD: shadow-7vm8 /usr/sbin/lctl get_param -n          mdd.lustre-MDT0000.lfsck_namespace |
      awk '/^status/ { print \$2 }'
CMD: shadow-7vm8 /usr/sbin/lctl get_param -n mdd.lustre-MDT0000.lfsck_namespace
'ls' should success after namespace LFSCK repairing
Resetting fail_loc on all nodes...CMD: shadow-7vm4,shadow-7vm6,shadow-7vm7,shadow-
7vm8,shadow-7vm9.shadow.whamcloud.com lctl set_param -n fail_loc=0          fail_val=0
2>/dev/null || true
done.
CMD: shadow-7vm4,shadow-7vm6,shadow-7vm7,shadow-7vm8 rc=\$([ -f /proc/sys/lnet/catastrophe ]
&&
      echo \$(< /proc/sys/lnet/catastrophe) || echo 0);
      if [ \$rc -ne 0 ]; then echo \$(hostname): \$rc; fi
      exit \$rc
PASS 23a (5s)

```

== sanity-lfsck test 23b: LFSCK can repair dangling name entry (2) == 06:33:45 (1409985225)

#####

The objectA has multiple hard links, one of them corresponding to the name entry\_B. But there is something wrong for the name entry\_B and cause entry\_B to references non-exist object\_C. In the first-stage scanning, the LFSCK will think the entry\_B as dangling, and re-create the lost object\_C. When the LFSCK comes to the second-stage scanning, it will find that the former re-creating object\_C is not proper, and will try to replace the object\_C with the real object\_A.

#####

Inject failure stub on MDT0 to simulate dangling name entry

```

CMD: shadow-7vm8 /usr/sbin/lctl set_param fail_loc=0x1621

```

```

fail_loc=0x1621

```

```

CMD: shadow-7vm8 /usr/sbin/lctl set_param fail_loc=0

```

```

fail_loc=0

```

'ls' should fail because of dangling name entry

Trigger namespace LFSCK to find out dangling name entry

```

CMD: shadow-7vm8 /usr/sbin/lctl lfsck_start -M lustre-MDT0000 -t namespace -r -C

```

```

Started LFSCK on the device lustre-MDT0000: scrub namespace

```

```

CMD: shadow-7vm8 /usr/sbin/lctl get_param -n          mdd.lustre-MDT0000.lfsck_namespace |
      awk '/^status/ { print \$2 }'

```

```

CMD: shadow-7vm8 /usr/sbin/lctl get_param -n          mdd.lustre-MDT0000.lfsck_namespace |
      awk '/^status/ { print \$2 }'

```

Waiting 32 secs for update

```

CMD: shadow-7vm8 /usr/sbin/lctl get_param -n          mdd.lustre-MDT0000.lfsck_namespace |
      awk '/^status/ { print \$2 }'

```

```

CMD: shadow-7vm8 /usr/sbin/lctl get_param -n          mdd.lustre-MDT0000.lfsck_namespace |
      awk '/^status/ { print \$2 }'

```

```

CMD: shadow-7vm8 /usr/sbin/lctl get_param -n          mdd.lustre-MDT0000.lfsck_namespace |

```

```

    awk '/^status/ { print \$2 }'
CMD: shadow-7vm8 /usr/sbin/lctl get_param -n          mdd.lustre-MDT0000.lfsck_namespace |
    awk '/^status/ { print \$2 }'
Updated after 4s: wanted 'completed' got 'completed'
CMD: shadow-7vm8 /usr/sbin/lctl get_param -n mdd.lustre-MDT0000.lfsck_namespace
CMD: shadow-7vm8 /usr/sbin/lctl get_param -n mdd.lustre-MDT0000.lfsck_namespace
Resetting fail_loc on all nodes...CMD: shadow-7vm4,shadow-7vm6,shadow-7vm7,shadow-
7vm8,shadow-7vm9.shadow.whamcloud.com lctl set_param -n fail_loc=0          fail_val=0
2>/dev/null || true
done.
CMD: shadow-7vm4,shadow-7vm6,shadow-7vm7,shadow-7vm8 rc=\$([ -f /proc/sys/lnet/catastrophe ]
&&
    echo \$(< /proc/sys/lnet/catastrophe) || echo 0);
    if [ \$?rc -ne 0 ]; then echo \$(hostname): \$?rc; fi
    exit \$?rc
PASS 23b (8s)

```

== sanity-lfsck test 23c: LFSCK can repair dangling name entry (3) == 06:33:54 (1409985234)

#####

The objectA has multiple hard links, one of them corresponding to the name entry\_B. But there is something wrong for the name entry\_B and cause entry\_B to references non-exist object\_C. In the first-stage scanning, the LFSCK will think the entry\_B as dangling, and re-create the lost object\_C. And then others modified the re-created object\_C. When the LFSCK comes to the second-stage scanning, it will find that the former re-creating object\_C maybe wrong and try to replace the object\_C with the real object\_A. But because object\_C has been modified, so the LFSCK cannot replace it.

#####

Inject failure stub on MDT0 to simulate dangling name entry

```
CMD: shadow-7vm8 /usr/sbin/lctl set_param fail_loc=0x1621
```

```
fail_loc=0x1621
```

```
CMD: shadow-7vm8 /usr/sbin/lctl set_param fail_loc=0
```

```
fail_loc=0
```

'ls' should fail because of dangling name entry

```
CMD: shadow-7vm8 /usr/sbin/lctl set_param fail_val=10 fail_loc=0x1602
```

```
fail_val=10
```

```
fail_loc=0x1602
```

Trigger namespace LFSCK to find out dangling name entry

```
CMD: shadow-7vm8 /usr/sbin/lctl lfsck_start -M lustre-MDT0000 -t namespace -r -C
```

Started LFSCK on the device lustre-MDT0000: scrub namespace

```
CMD: shadow-7vm8 /usr/sbin/lctl set_param fail_val=0 fail_loc=0
```

```
fail_val=0
```

```
fail_loc=0
```

```
CMD: shadow-7vm8 /usr/sbin/lctl get_param -n          mdd.lustre-MDT0000.lfsck_namespace |
```

```
    awk '/^status/ { print \$2 }'
```

```
CMD: shadow-7vm8 /usr/sbin/lctl get_param -n          mdd.lustre-MDT0000.lfsck_namespace |
```

```
    awk '/^status/ { print \$2 }'
```



Waiting 32 secs for update

```
CMD: shadow-7vm8 /usr/sbin/lctl get_param -n mdd.lustre-MDT0000.lfsck_namespace |
      awk '/^status/ { print \$2 }'
CMD: shadow-7vm8 /usr/sbin/lctl get_param -n mdd.lustre-MDT0000.lfsck_namespace |
      awk '/^status/ { print \$2 }'
CMD: shadow-7vm8 /usr/sbin/lctl get_param -n mdd.lustre-MDT0000.lfsck_namespace |
      awk '/^status/ { print \$2 }'
CMD: shadow-7vm8 /usr/sbin/lctl get_param -n mdd.lustre-MDT0000.lfsck_namespace |
      awk '/^status/ { print \$2 }'
CMD: shadow-7vm8 /usr/sbin/lctl get_param -n mdd.lustre-MDT0000.lfsck_namespace |
      awk '/^status/ { print \$2 }'
CMD: shadow-7vm8 /usr/sbin/lctl get_param -n mdd.lustre-MDT0000.lfsck_namespace |
      awk '/^status/ { print \$2 }'
CMD: shadow-7vm8 /usr/sbin/lctl get_param -n mdd.lustre-MDT0000.lfsck_namespace |
      awk '/^status/ { print \$2 }'
```

Updated after 7s: wanted 'completed' got 'completed'

```
CMD: shadow-7vm8 /usr/sbin/lctl get_param -n mdd.lustre-MDT0000.lfsck_namespace
Resetting fail_loc on all nodes...CMD: shadow-7vm4,shadow-7vm6,shadow-7vm7,shadow-
7vm8,shadow-7vm9.shadow.whamcloud.com lctl set_param -n fail_loc=0 fail_val=0
2>/dev/null || true
```

done.

```
CMD: shadow-7vm4,shadow-7vm6,shadow-7vm7,shadow-7vm8 rc=\$([ -f /proc/sys/lnet/catastrophe ]
&&
```

```
    echo \$(< /proc/sys/lnet/catastrophe) || echo 0);
    if [ \$?rc -ne 0 ]; then echo \$(hostname): \$?rc; fi
    exit \$?rc
```

PASS 23c (15s)

== sanity-lfsck test 24: LFSCK can repair multiple-referenced name entry == 06:34:09  
(1409985249)

#####

Two MDT-objects back reference the same name entry via their each own linkEA entry, but the name entry only references one MDT-object. The namespace LFSCK will remove the linkEA entry for the MDT-object that is not recognized. If such MDT-object has no other linkEA entry after the removing, then the LFSCK will add it as orphan under the .lustre/lost+found/MDTxxxx/.

#####

```
[0x340000bd1:0x7:0x0]
```

```
[0x340000bd1:0x8:0x0]
```

Inject failure stub on MDT0 to simulate the case that the /mnt/lustre/d24.sanity-lfsck/d0/dummy/foo has the 'bad' linkEA entry that references /mnt/lustre/d24.sanity-lfsck/d0/guard/foo.

Then remove the name entry /mnt/lustre/d24.sanity-lfsck/d0/dummy/foo.

So the MDT-object /mnt/lustre/d24.sanity-lfsck/d0/dummy/foo will be left there with the same linkEA entry as another MDT-object

/mnt/lustre/d24.sanity-lfsck/d0/guard/foo has

```
CMD: shadow-7vm8 /usr/sbin/lctl set_param fail_loc=0x1622
```

```
fail_loc=0x1622
```

```

CMD: shadow-7vm8 /usr/sbin/lctl set_param fail_loc=0
fail_loc=0
stat /mnt/lustre/d24.sanity-lfsck/d0/dummy/foo should fail
Trigger namespace LFSCK to repair multiple-referenced name entry
CMD: shadow-7vm8 /usr/sbin/lctl lfsck_start -M lustre-MDT0000 -t namespace -A -r
Started LFSCK on the device lustre-MDT0000: scrub namespace
CMD: shadow-7vm8 /usr/sbin/lctl get_param -n          mdd.lustre-MDT0000.lfsck_namespace |
      awk '/^status/ { print \$2 }'
CMD: shadow-7vm8 /usr/sbin/lctl get_param -n          mdd.lustre-MDT0000.lfsck_namespace |
      awk '/^status/ { print \$2 }'
CMD: shadow-7vm8 /usr/sbin/lctl get_param -n mdd.lustre-MDT0000.lfsck_namespace
There should be an orphan under .lustre/lost+found/MDT0000/
Resetting fail_loc on all nodes...CMD: shadow-7vm4,shadow-7vm6,shadow-7vm7,shadow-
7vm8,shadow-7vm9.shadow.whamcloud.com lctl set_param -n fail_loc=0          fail_val=0
2>/dev/null || true
done.
CMD: shadow-7vm4,shadow-7vm6,shadow-7vm7,shadow-7vm8 rc=\$( [ -f /proc/sys/lnet/catastrophe ]
&&
      echo \$(< /proc/sys/lnet/catastrophe) || echo 0);
      if [ \$?rc -ne 0 ]; then echo \$(hostname): \$?rc; fi
      exit \$?rc
PASS 24 (3s)

```

```

== sanity-lfsck test 25: LFSCK can repair bad file type in the name entry == 06:34:12
(1409985252)

```

```
#####
```

The file type in the name entry does not match the file type claimed by the referenced object. Then the LFSCK will update the file type in the name entry.

```
#####
```

Inject failure stub on MDT0 to simulate the case that the file type stored in the name entry is wrong.

```

CMD: shadow-7vm8 /usr/sbin/lctl set_param fail_loc=0x1623
fail_loc=0x1623
CMD: shadow-7vm8 /usr/sbin/lctl set_param fail_loc=0
fail_loc=0
Trigger namespace LFSCK to repair bad file type in the name entry
CMD: shadow-7vm8 /usr/sbin/lctl lfsck_start -M lustre-MDT0000 -t namespace -r
Started LFSCK on the device lustre-MDT0000: scrub namespace
CMD: shadow-7vm8 /usr/sbin/lctl get_param -n          mdd.lustre-MDT0000.lfsck_namespace |
      awk '/^status/ { print \$2 }'
CMD: shadow-7vm8 /usr/sbin/lctl get_param -n          mdd.lustre-MDT0000.lfsck_namespace |
      awk '/^status/ { print \$2 }'
CMD: shadow-7vm8 /usr/sbin/lctl get_param -n mdd.lustre-MDT0000.lfsck_namespace
total 8
216172967082590341 drwxr-xr-x 2 root root 4096 Sep  6 06:34 .
216172967082590340 drwxr-xr-x 3 root root 4096 Sep  6 06:34 ..
216172967082590342 -rw-r--r-- 1 root root    0 Sep  6 06:34 foo

```

```

Resetting fail_loc on all nodes...CMD: shadow-7vm4,shadow-7vm6,shadow-7vm7,shadow-
7vm8,shadow-7vm9.shadow.whamcloud.com lctl set_param -n fail_loc=0          fail_val=0
2>/dev/null || true
done.
CMD: shadow-7vm4,shadow-7vm6,shadow-7vm7,shadow-7vm8 rc=\$([ -f /proc/sys/lnet/catastrophe ]
&&
    echo \$(< /proc/sys/lnet/catastrophe) || echo 0);
    if [ \$rc -ne 0 ]; then echo \$(hostname): \$rc; fi
    exit \$rc
PASS 25 (3s)

== sanity-lfsck test 26a: LFSCK can add the missed local name entry back to the namespace ==
06:34:15 (1409985255)
#####
The local name entry back referenced by the MDT-object is lost.
The namespace LFSCK will add the missed local name entry back
to the normal namespace.
#####
Inject failure stub on MDT0 to simulate the case that
foo's name entry will be removed, but the foo's object
and its linkEA are kept in the system.
CMD: shadow-7vm8 /usr/sbin/lctl set_param fail_loc=0x1624
fail_loc=0x1624
CMD: shadow-7vm8 /usr/sbin/lctl set_param fail_loc=0
fail_loc=0
Trigger namespace LFSCK to repair the missed remote name entry
CMD: shadow-7vm8 /usr/sbin/lctl lfsck_start -M lustre-MDT0000 -t namespace -r -A
Started LFSCK on the device lustre-MDT0000: scrub namespace
CMD: shadow-7vm8 /usr/sbin/lctl get_param -n          mdd.lustre-MDT0000.lfsck_namespace |
    awk '/^status/ { print \$2 }'
CMD: shadow-7vm8 /usr/sbin/lctl get_param -n          mdd.lustre-MDT0000.lfsck_namespace |
    awk '/^status/ { print \$2 }'
Waiting 32 secs for update
CMD: shadow-7vm8 /usr/sbin/lctl get_param -n          mdd.lustre-MDT0000.lfsck_namespace |
    awk '/^status/ { print \$2 }'
CMD: shadow-7vm8 /usr/sbin/lctl get_param -n          mdd.lustre-MDT0000.lfsck_namespace |
    awk '/^status/ { print \$2 }'
Updated after 2s: wanted 'completed' got 'completed'
CMD: shadow-7vm8 /usr/sbin/lctl get_param -n mdd.lustre-MDT0000.lfsck_namespace
216172967082590345 -rw-r--r-- 2 root root 0 Sep  6 06:34 /mnt/lustre/d26a.sanity-
-lfsck/d0/foo
Resetting fail_loc on all nodes...CMD: shadow-7vm4,shadow-7vm6,shadow-7vm7,shadow-
7vm8,shadow-7vm9.shadow.whamcloud.com lctl set_param -n fail_loc=0          fail_val=0
2>/dev/null || true
done.
CMD: shadow-7vm4,shadow-7vm6,shadow-7vm7,shadow-7vm8 rc=\$([ -f /proc/sys/lnet/catastrophe ]
&&
    echo \$(< /proc/sys/lnet/catastrophe) || echo 0);
    if [ \$rc -ne 0 ]; then echo \$(hostname): \$rc; fi
    exit \$rc

```

PASS 26a (5s)

== sanity-lfsck test 26b: LFSCK can add the missed remote name entry back to the namespace == 06:34:21 (1409985261)

#####

The remote name entry back referenced by the MDT-object is lost.  
The namespace LFSCK will add the missed remote name entry back to the normal namespace.

#####

Inject failure stub on MDT0 to simulate the case that foo's name entry will be removed, but the foo's object and its linkEA are kept in the system.

```
CMD: shadow-7vm8 /usr/sbin/lctl set_param fail_loc=0x1624
fail_loc=0x1624
```

```
CMD: shadow-7vm8 /usr/sbin/lctl set_param fail_loc=0
fail_loc=0
```

Trigger namespace LFSCK to repair the missed remote name entry

```
CMD: shadow-7vm8 /usr/sbin/lctl lfsck_start -M lustre-MDT0000 -t namespace -r -A
Started LFSCK on the device lustre-MDT0000: scrub namespace
```

```
CMD: shadow-7vm8 /usr/sbin/lctl get_param -n mdd.lustre-MDT0000.lfsck_namespace |
awk '/^status/ { print \$2 }'
```

```
CMD: shadow-7vm8 /usr/sbin/lctl get_param -n mdd.lustre-MDT0000.lfsck_namespace |
awk '/^status/ { print \$2 }'
```

```
CMD: shadow-7vm8 /usr/sbin/lctl get_param -n mdd.lustre-MDT0000.lfsck_namespace
total 8
```

```
216172967082590347 drwxr-xr-x 2 root root 4096 Sep  6 06:34 .
```

```
234187231374344202 drwxr-xr-x 3 root root 4096 Sep  6 06:34 ..
```

```
Resetting fail_loc on all nodes...CMD: shadow-7vm4,shadow-7vm6,shadow-7vm7,shadow-
7vm8,shadow-7vm9.shadow.whamcloud.com lctl set_param -n fail_loc=0 fail_val=0
2>/dev/null || true
```

done.

```
CMD: shadow-7vm4,shadow-7vm6,shadow-7vm7,shadow-7vm8 rc=\${[ -f /proc/sys/lnet/catastrophe ]
&&
```

```
    echo \$(< /proc/sys/lnet/catastrophe) || echo 0);
    if [ \$?rc -ne 0 ]; then echo \$(hostname): \$?rc; fi
    exit $?rc
```

PASS 26b (3s)

== sanity-lfsck test 27a: LFSCK can recreate the lost local parent directory as orphan == 06:34:24 (1409985264)

#####

The local parent referenced by the MDT-object linkEA is lost.  
The namespace LFSCK will re-create the lost parent as orphan.

#####

Inject failure stub on MDT0 to simulate the case that foo's name entry will be removed, but the foo's object and its linkEA are kept in the system. And then remove another hard link and the parent directory.

```
CMD: shadow-7vm8 /usr/sbin/lctl set_param fail_loc=0x1624
fail_loc=0x1624
```

```

CMD: shadow-7vm8 /usr/sbin/lctl set_param fail_loc=0
fail_loc=0
Trigger namespace LFSCK to repair the lost parent
CMD: shadow-7vm8 /usr/sbin/lctl lfsck_start -M lustre-MDT0000 -t namespace -r -A
Started LFSCK on the device lustre-MDT0000: scrub namespace
CMD: shadow-7vm8 /usr/sbin/lctl get_param -n          mdd.lustre-MDT0000.lfsck_namespace |
      awk '/^status/ { print \$2 }'
CMD: shadow-7vm8 /usr/sbin/lctl get_param -n          mdd.lustre-MDT0000.lfsck_namespace |
      awk '/^status/ { print \$2 }'
CMD: shadow-7vm8 /usr/sbin/lctl get_param -n mdd.lustre-MDT0000.lfsck_namespace
There should be an orphan under .lustre/lost+found/MDT0000/
total 12
216172899973726209 drwx----- 4 root root 4096 Sep  6 06:34 .
144115188109410307 dr-x----- 4 root root 4096 Sep  6 06:19 ..
216172967082590349 drwx----- 3 root root 4096 Jan  1 1970 [0x300002b11:0x8d:0x0]-P-0
Resetting fail_loc on all nodes...CMD: shadow-7vm4,shadow-7vm6,shadow-7vm7,shadow-
7vm8,shadow-7vm9.shadow.whamcloud.com lctl set_param -n fail_loc=0          fail_val=0
2>/dev/null || true
done.
CMD: shadow-7vm4,shadow-7vm6,shadow-7vm7,shadow-7vm8 rc=\$([ -f /proc/sys/lnet/catastrophe ]
&&
      echo \$(< /proc/sys/lnet/catastrophe) || echo 0);
      if [ \$rc -ne 0 ]; then echo \$(hostname): \$rc; fi
      exit \$rc

```

PASS 27a (3s)

== sanity-lfsck test 27b: LFSCK can recreate the lost remote parent directory as orphan ==  
06:34:27 (1409985267)

#####

The remote parent referenced by the MDT-object linKEA is lost.  
The namespace LFSCK will re-create the lost parent as orphan.

#####

[0x340000bd1:0xb:0x0]

Inject failure stub on MDT0 to simulate the case that  
foo's name entry will be removed, but the foo's object  
and its linKEA are kept in the system. And then remove  
the parent directory.

```

CMD: shadow-7vm8 /usr/sbin/lctl set_param fail_loc=0x1624

```

```

fail_loc=0x1624

```

```

CMD: shadow-7vm8 /usr/sbin/lctl set_param fail_loc=0

```

```

fail_loc=0

```

Trigger namespace LFSCK to repair the missed remote name entry

```

CMD: shadow-7vm8 /usr/sbin/lctl lfsck_start -M lustre-MDT0000 -t namespace -r -A

```

Started LFSCK on the device lustre-MDT0000: scrub namespace

```

CMD: shadow-7vm8 /usr/sbin/lctl get_param -n          mdd.lustre-MDT0000.lfsck_namespace |
      awk '/^status/ { print \$2 }'

```

```

CMD: shadow-7vm8 /usr/sbin/lctl get_param -n          mdd.lustre-MDT0000.lfsck_namespace |
      awk '/^status/ { print \$2 }'

```

```

CMD: shadow-7vm8 /usr/sbin/lctl get_param -n mdd.lustre-MDT0000.lfsck_namespace

```

```

total 12
144115188109410307 dr-x----- 4 root root 4096 Sep  6 06:19 .
216172899973726209 drwx----- 3 root root 4096 Sep  6 06:34 MDT0000
234187197819912193 drwx----- 4 root root 4096 Sep  6 06:33 MDT0001
There should be an orphan under .lustre/lost+found/MDT0001/
total 12
234187197819912193 drwx----- 4 root root 4096 Sep  6 06:33 .
144115188109410307 dr-x----- 4 root root 4096 Sep  6 06:19 ..
234187231374344203 drwx----- 4 root root 4096 Jan  1 1970 [0x340000bd1:0xb:0x0]-P-0
Resetting fail_loc on all nodes...CMD: shadow-7vm4,shadow-7vm6,shadow-7vm7,shadow-
7vm8,shadow-7vm9.shadow.whamcloud.com lctl set_param -n fail_loc=0          fail_val=0
2>/dev/null || true
done.
CMD: shadow-7vm4,shadow-7vm6,shadow-7vm7,shadow-7vm8 rc=\$([ -f /proc/sys/lnet/catastrophe ]
&&
    echo \$(< /proc/sys/lnet/catastrophe) || echo 0);
    if [ \$rc -ne 0 ]; then echo \$(hostname): \$rc; fi
    exit \$rc
PASS 27b (4s)

== sanity-lfscck test 29a: LFSCK can repair bad nlink count (1) == 06:34:39 (1409985279)
#####
The object's nlink attribute is larger than the object's known
name entries count. The LFSCK will repair the object's nlink
attribute to match the known name entries count
#####
rm: cannot remove `mnt/lustre/.lustre/lost+found/MDT0001/[0x340000bd1:0xb:0x0]-P-0':
Directory not empty
Inject failure stub on MDT0 to simulate the case that foo's
nlink attribute is larger than its name entries count.
CMD: shadow-7vm8 /usr/sbin/lctl set_param fail_loc=0x1625
fail_loc=0x1625
CMD: shadow-7vm8 /usr/sbin/lctl set_param fail_loc=0
fail_loc=0
Trigger namespace LFSCK to repair the nlink count
CMD: shadow-7vm8 /usr/sbin/lctl lfscck_start -M lustre-MDT0000 -t namespace -r -A
Started LFSCK on the device lustre-MDT0000: scrub namespace
CMD: shadow-7vm8 /usr/sbin/lctl get_param -n          mdd.lustre-MDT0000.lfscck_namespace |
    awk '/^status/ { print \$2 }'
CMD: shadow-7vm8 /usr/sbin/lctl get_param -n          mdd.lustre-MDT0000.lfscck_namespace |
    awk '/^status/ { print \$2 }'
CMD: shadow-7vm8 /usr/sbin/lctl get_param -n mdd.lustre-MDT0000.lfscck_namespace
Resetting fail_loc on all nodes...CMD: shadow-7vm4,shadow-7vm6,shadow-7vm7,shadow-
7vm8,shadow-7vm9.shadow.whamcloud.com lctl set_param -n fail_loc=0          fail_val=0
2>/dev/null || true
done.
CMD: shadow-7vm4,shadow-7vm6,shadow-7vm7,shadow-7vm8 rc=\$([ -f /proc/sys/lnet/catastrophe ]
&&
    echo \$(< /proc/sys/lnet/catastrophe) || echo 0);
    if [ \$rc -ne 0 ]; then echo \$(hostname): \$rc; fi

```

```
exit \src
```

PASS 29a (3s)

```
== sanity-lfsck test 29b: LFSCK can repair bad nlink count (2) == 06:34:42 (1409985282)
```

```
#####
```

The object's nlink attribute is smaller than the object's known name entries count. The LFSCK will repair the object's nlink attribute to match the known name entries count

```
#####
```

```
rm: cannot remove `/mnt/lustre/.lustre/lost+found/MDT0001/[0x34000bd1:0xb:0x0]-P-0':  
Directory not empty
```

Inject failure stub on MDT0 to simulate the case that foo's nlink attribute is smaller than its name entries count.

```
CMD: shadow-7vm8 /usr/sbin/lctl set_param fail_loc=0x1626
```

```
fail_loc=0x1626
```

```
CMD: shadow-7vm8 /usr/sbin/lctl set_param fail_loc=0
```

```
fail_loc=0
```

Trigger namespace LFSCK to repair the nlink count

```
CMD: shadow-7vm8 /usr/sbin/lctl lfsck_start -M lustre-MDT0000 -t namespace -r -A
```

Started LFSCK on the device lustre-MDT0000: scrub namespace

```
CMD: shadow-7vm8 /usr/sbin/lctl get_param -n mdd.lustre-MDT0000.lfsck_namespace |  
awk '/^status/ { print \ $2 }'
```

```
CMD: shadow-7vm8 /usr/sbin/lctl get_param -n mdd.lustre-MDT0000.lfsck_namespace |  
awk '/^status/ { print \ $2 }'
```

Waiting 32 secs for update

```
CMD: shadow-7vm8 /usr/sbin/lctl get_param -n mdd.lustre-MDT0000.lfsck_namespace |  
awk '/^status/ { print \ $2 }'
```

```
CMD: shadow-7vm8 /usr/sbin/lctl get_param -n mdd.lustre-MDT0000.lfsck_namespace |  
awk '/^status/ { print \ $2 }'
```

```
CMD: shadow-7vm8 /usr/sbin/lctl get_param -n mdd.lustre-MDT0000.lfsck_namespace |  
awk '/^status/ { print \ $2 }'
```

Updated after 3s: wanted 'completed' got 'completed'

```
CMD: shadow-7vm8 /usr/sbin/lctl get_param -n mdd.lustre-MDT0000.lfsck_namespace
```

```
Resetting fail_loc on all nodes...CMD: shadow-7vm4,shadow-7vm6,shadow-7vm7,shadow-  
7vm8,shadow-7vm9.shadow.whamcloud.com lctl set_param -n fail_loc=0 fail_val=0  
2>/dev/null || true
```

done.

```
CMD: shadow-7vm4,shadow-7vm6,shadow-7vm7,shadow-7vm8 rc=\$( [ -f /proc/sys/lnet/catastrophe ]  
&&
```

```
echo \$(< /proc/sys/lnet/catastrophe) || echo 0);  
if [ \src -ne 0 ]; then echo \$(hostname): \src; fi  
exit \src
```

PASS 29b (7s)

```
== sanity-lfsck test 29c: Not verify nlink attr if hark links exceed linkEA limitation ==  
06:34:49 (1409985289)
```

```
#####
```

There are too much hard links to the object, and exceeds the echo object's linkEA limitation, as to NOT all the known name entries will be recorded in the linkEA. Under such case, LFSCK should

skip the nlink verification for this object.

#####

rm: cannot remove `/mnt/lustre/.lustre/lost+found/MDT0001/[0x340000bd1:0xb:0x0]-P-0':  
Directory not empty

Inject failure stub on MDT0 to simulate the case that  
foo's hard links exceed the object's linKEA limitation.

CMD: shadow-7vm8 /usr/sbin/lctl set\_param fail\_loc=0x1627  
fail\_loc=0x1627

/mnt/lustre/d29c.sanity-lfsck/d0/foo

/mnt/lustre/d29c.sanity-lfsck/d0/h1

Trigger namespace LFSCK to repair the nlink count

CMD: shadow-7vm8 /usr/sbin/lctl lfsck\_start -M lustre-MDT0000 -t namespace -r -A

Started LFSCK on the device lustre-MDT0000: scrub namespace

CMD: shadow-7vm8 /usr/sbin/lctl get\_param -n mdd.lustre-MDT0000.lfsck\_namespace |  
awk '/^status/ { print \$2 }'

CMD: shadow-7vm8 /usr/sbin/lctl get\_param -n mdd.lustre-MDT0000.lfsck\_namespace |  
awk '/^status/ { print \$2 }'

Waiting 32 secs for update

CMD: shadow-7vm8 /usr/sbin/lctl get\_param -n mdd.lustre-MDT0000.lfsck\_namespace |  
awk '/^status/ { print \$2 }'

CMD: shadow-7vm8 /usr/sbin/lctl get\_param -n mdd.lustre-MDT0000.lfsck\_namespace |  
awk '/^status/ { print \$2 }'

Updated after 2s: wanted 'completed' got 'completed'

CMD: shadow-7vm8 /usr/sbin/lctl set\_param fail\_loc=0  
fail\_loc=0

CMD: shadow-7vm8 /usr/sbin/lctl get\_param -n mdd.lustre-MDT0000.lfsck\_namespace

Resetting fail\_loc on all nodes...CMD: shadow-7vm4,shadow-7vm6,shadow-7vm7,shadow-  
7vm8,shadow-7vm9.shadow.whamcloud.com lctl set\_param -n fail\_loc=0 fail\_val=0  
2>/dev/null || true

done.

CMD: shadow-7vm4,shadow-7vm6,shadow-7vm7,shadow-7vm8 rc=\$(( [ -f /proc/sys/lnet/catastrophe ]  
&&

```
echo \$(< /proc/sys/lnet/catastrophe) || echo 0);  
if [ \${rc} -ne 0 ]; then echo \$(hostname): \${rc}; fi  
exit \${rc}
```

PASS 29c (6s)

== sanity-lfsck test 31a: The LFSCK can find/repair the name entry with bad name hash (1) ==  
06:35:24 (1409985324)

#####

For the name entry under a striped directory, if the name  
hash does not match the shard, then the LFSCK will repair  
the bad name entry

#####

rm: cannot remove `/mnt/lustre/.lustre/lost+found/MDT0001/[0x340000bd1:0xb:0x0]-P-0':  
Directory not empty

Inject failure stub on client to simulate the case that  
some name entry should be inserted into the second shard  
but inserted into the first shard by wrong

fail\_loc=0x1628



```
fail_val=0
total: 4 creates in 0.04 seconds: 110.67 creates/second
fail_loc=0
fail_val=0
Trigger namespace LFSCK to repair bad name hash
CMD: shadow-7vm8 /usr/sbin/lctl lfsck_start -M lustre-MDT0000 -t namespace -r -A
Started LFSCK on the device lustre-MDT0000: scrub namespace
CMD: shadow-7vm8 /usr/sbin/lctl get_param -n          mdd.lustre-MDT0000.lfsck_namespace |
      awk '/^status/ { print \$2 }'
CMD: shadow-7vm8 /usr/sbin/lctl get_param -n          mdd.lustre-MDT0000.lfsck_namespace |
      awk '/^status/ { print \$2 }'
CMD: shadow-7vm8 /usr/sbin/lctl get_param -n mdd.lustre-MDT0000.lfsck_namespace
10.1.4.75@tcp:/lustre /mnt/lustre lustre rw,flock,user_xattr 0 0
CMD: shadow-7vm9.shadow.whamcloud.com grep -c /mnt/lustre' ' /proc/mounts
Stopping client shadow-7vm9.shadow.whamcloud.com /mnt/lustre (opts:)
CMD: shadow-7vm9.shadow.whamcloud.com lsof -t /mnt/lustre
CMD: shadow-7vm9.shadow.whamcloud.com umount /mnt/lustre 2>&1
Starting client: shadow-7vm9.shadow.whamcloud.com: -o user_xattr,flock shadow-
7vm8@tcp:/lustre /mnt/lustre
CMD: shadow-7vm9.shadow.whamcloud.com mkdir -p /mnt/lustre
CMD: shadow-7vm9.shadow.whamcloud.com mount -t lustre -o user_xattr,flock shadow-
7vm8@tcp:/lustre /mnt/lustre
  File: `/mnt/lustre/d31a.sanity-lfsck/striped_dir/d0'
  Size: 4096          Blocks: 8          IO Block: 4096  directory
Device: 2c54f966h/743766374d      Inode: 216173000620244995  Links: 2
Access: (0755/drwxr-xr-x)  Uid: (  0/   root)   Gid: (  0/   root)
Access: 2014-09-06 06:35:25.000000000 +0000
Modify: 2014-09-06 06:35:25.000000000 +0000
Change: 2014-09-06 06:35:25.000000000 +0000
  File: `/mnt/lustre/d31a.sanity-lfsck/striped_dir/d1'
  Size: 4096          Blocks: 8          IO Block: 4096  directory
Device: 2c54f966h/743766374d      Inode: 216173000620244996  Links: 2
Access: (0755/drwxr-xr-x)  Uid: (  0/   root)   Gid: (  0/   root)
Access: 2014-09-06 06:35:25.000000000 +0000
Modify: 2014-09-06 06:35:25.000000000 +0000
Change: 2014-09-06 06:35:25.000000000 +0000
  File: `/mnt/lustre/d31a.sanity-lfsck/striped_dir/d2'
  Size: 4096          Blocks: 8          IO Block: 4096  directory
Device: 2c54f966h/743766374d      Inode: 216173000620244997  Links: 2
Access: (0755/drwxr-xr-x)  Uid: (  0/   root)   Gid: (  0/   root)
Access: 2014-09-06 06:35:25.000000000 +0000
Modify: 2014-09-06 06:35:25.000000000 +0000
Change: 2014-09-06 06:35:25.000000000 +0000
  File: `/mnt/lustre/d31a.sanity-lfsck/striped_dir/d3'
  Size: 4096          Blocks: 8          IO Block: 4096  directory
Device: 2c54f966h/743766374d      Inode: 216173000620244998  Links: 2
Access: (0755/drwxr-xr-x)  Uid: (  0/   root)   Gid: (  0/   root)
Access: 2014-09-06 06:35:25.000000000 +0000
Modify: 2014-09-06 06:35:25.000000000 +0000
```

Change: 2014-09-06 06:35:25.000000000 +0000

Resetting fail\_loc on all nodes...CMD: shadow-7vm4,shadow-7vm6,shadow-7vm7,shadow-7vm8,shadow-7vm9.shadow.whamcloud.com lctl set\_param -n fail\_loc=0 fail\_val=0 2>/dev/null || true

done.

CMD: shadow-7vm4,shadow-7vm6,shadow-7vm7,shadow-7vm8 rc=\\${[ -f /proc/sys/lnet/catastrophe ] &&

```
    echo \${(< /proc/sys/lnet/catastrophe) || echo 0};
    if [ \${rc} -ne 0 ]; then echo \$(hostname): \${rc}; fi
    exit \${rc}
```

PASS 31a (4s)

== sanity-lfsck test 31b: The LFSCK can find/repair the name entry with bad name hash (2) ==  
06:35:28 (1409985328)

#####

For the name entry under a striped directory, if the name hash does not match the shard, then the LFSCK will repair the bad name entry

#####

rm: cannot remove `/mnt/lustre/.lustre/lost+found/MDT0001/[0x340000bd1:0xb:0x0]-P-0':  
Directory not empty

Inject failure stub on client to simulate the case that some name entry should be inserted into the first shard but inserted into the second shard by wrong

fail\_loc=0x1628

fail\_val=1

total: 4 creates in 0.02 seconds: 237.83 creates/second

fail\_loc=0

fail\_val=0

Trigger namespace LFSCK to repair bad name hash

CMD: shadow-7vm8 /usr/sbin/lctl lfsck\_start -M lustre-MDT0000 -t namespace -r -A

Started LFSCK on the device lustre-MDT0000: scrub namespace

CMD: shadow-7vm4 /usr/sbin/lctl get\_param -n mdd.lustre-MDT0001.lfsck\_namespace |  
awk '/^status/ { print \\$2 }'

CMD: shadow-7vm4 /usr/sbin/lctl get\_param -n mdd.lustre-MDT0001.lfsck\_namespace |  
awk '/^status/ { print \\$2 }'

CMD: shadow-7vm4 /usr/sbin/lctl get\_param -n mdd.lustre-MDT0001.lfsck\_namespace  
10.1.4.75@tcp:/lustre /mnt/lustre lustre rw,flock,user\_xattr 0 0

CMD: shadow-7vm9.shadow.whamcloud.com grep -c /mnt/lustre' ' /proc/mounts

Stopping client shadow-7vm9.shadow.whamcloud.com /mnt/lustre (opts:)

CMD: shadow-7vm9.shadow.whamcloud.com lsof -t /mnt/lustre

CMD: shadow-7vm9.shadow.whamcloud.com umount /mnt/lustre 2>&1

Starting client: shadow-7vm9.shadow.whamcloud.com: -o user\_xattr,flock shadow-7vm8@tcp:/lustre /mnt/lustre

CMD: shadow-7vm9.shadow.whamcloud.com mkdir -p /mnt/lustre

CMD: shadow-7vm9.shadow.whamcloud.com mount -t lustre -o user\_xattr,flock shadow-7vm8@tcp:/lustre /mnt/lustre

File: `/mnt/lustre/d31b.sanity-lfsck/striped\_dir/d0'

Size: 4096 Blocks: 8 IO Block: 4096 directory

Device: 2c54f966h/743766374d Inode: 234187231407898625 Links: 2

```

Access: (0755/drwxr-xr-x) Uid: ( 0/ root) Gid: ( 0/ root)
Access: 2014-09-06 06:35:29.000000000 +0000
Modify: 2014-09-06 06:35:29.000000000 +0000
Change: 2014-09-06 06:35:29.000000000 +0000
  File: `/mnt/lustre/d31b.sanity-lfsck/stripped_dir/d1'
  Size: 4096      Blocks: 8      IO Block: 4096  directory
Device: 2c54f966h/743766374d      Inode: 234187231407898626 Links: 2
Access: (0755/drwxr-xr-x) Uid: ( 0/ root) Gid: ( 0/ root)
Access: 2014-09-06 06:35:29.000000000 +0000
Modify: 2014-09-06 06:35:29.000000000 +0000
Change: 2014-09-06 06:35:29.000000000 +0000
  File: `/mnt/lustre/d31b.sanity-lfsck/stripped_dir/d2'
  Size: 4096      Blocks: 8      IO Block: 4096  directory
Device: 2c54f966h/743766374d      Inode: 234187231407898627 Links: 2
Access: (0755/drwxr-xr-x) Uid: ( 0/ root) Gid: ( 0/ root)
Access: 2014-09-06 06:35:29.000000000 +0000
Modify: 2014-09-06 06:35:29.000000000 +0000
Change: 2014-09-06 06:35:29.000000000 +0000
  File: `/mnt/lustre/d31b.sanity-lfsck/stripped_dir/d3'
  Size: 4096      Blocks: 8      IO Block: 4096  directory
Device: 2c54f966h/743766374d      Inode: 234187231407898628 Links: 2
Access: (0755/drwxr-xr-x) Uid: ( 0/ root) Gid: ( 0/ root)
Access: 2014-09-06 06:35:29.000000000 +0000
Modify: 2014-09-06 06:35:29.000000000 +0000
Change: 2014-09-06 06:35:29.000000000 +0000
Resetting fail_loc on all nodes...CMD: shadow-7vm4,shadow-7vm6,shadow-7vm7,shadow-
7vm8,shadow-7vm9.shadow.whamcloud.com lctl set_param -n fail_loc=0      fail_val=0
2>/dev/null || true
done.
CMD: shadow-7vm4,shadow-7vm6,shadow-7vm7,shadow-7vm8 rc=\$([ -f /proc/sys/lnet/catastrophe ]
&&
    echo \$(< /proc/sys/lnet/catastrophe) || echo 0);
    if [ \$?rc -ne 0 ]; then echo \$(hostname): \$?rc; fi
    exit \$?rc

```

PASS 31b (3s)

```

== sanity-lfsck test 31c: Re-generate the lost master LMV EA for striped directory ==
06:35:32 (1409985332)

```

#####

For some reason, the master MDT-object of the striped directory may lost its master LMV EA. If nobody created files under the master directly after the master LMV EA lost, then the LFSCK should re-generate the master LMV EA.

#####

```

rm: cannot remove `/mnt/lustre/.lustre/lost+found/MDT0001/[0x340000bd1:0xb:0x0]-P-0':
Directory not empty

```

Inject failure stub on MDT0 to simulate the case that the master MDT-object of the striped directory lost the LMV EA.

```

CMD: shadow-7vm8 /usr/sbin/lctl set_param fail_loc=0x1629
fail_loc=0x1629

```

```

CMD: shadow-7vm8 /usr/sbin/lctl set_param fail_loc=0
fail_loc=0
Trigger namespace LFSCK to re-generate master LMV EA
CMD: shadow-7vm8 /usr/sbin/lctl lfsck_start -M lustre-MDT0000 -t namespace -r -A
Started LFSCK on the device lustre-MDT0000: scrub namespace
CMD: shadow-7vm8 /usr/sbin/lctl get_param -n          mdd.lustre-MDT0000.lfsck_namespace |
      awk '/^status/ { print \$2 }'
CMD: shadow-7vm8 /usr/sbin/lctl get_param -n          mdd.lustre-MDT0000.lfsck_namespace |
      awk '/^status/ { print \$2 }'
CMD: shadow-7vm8 /usr/sbin/lctl get_param -n mdd.lustre-MDT0000.lfsck_namespace
10.1.4.75@tcp:/lustre /mnt/lustre lustre rw,flock,user_xattr 0 0
CMD: shadow-7vm9.shadow.whamcloud.com grep -c /mnt/lustre' ' /proc/mounts
Stopping client shadow-7vm9.shadow.whamcloud.com /mnt/lustre (opts:)
CMD: shadow-7vm9.shadow.whamcloud.com lsof -t /mnt/lustre
CMD: shadow-7vm9.shadow.whamcloud.com umount /mnt/lustre 2>&1
Starting client: shadow-7vm9.shadow.whamcloud.com: -o user_xattr,flock shadow-
7vm8@tcp:/lustre /mnt/lustre
CMD: shadow-7vm9.shadow.whamcloud.com mkdir -p /mnt/lustre
CMD: shadow-7vm9.shadow.whamcloud.com mount -t lustre -o user_xattr,flock shadow-
7vm8@tcp:/lustre /mnt/lustre
Resetting fail_loc on all nodes...CMD: shadow-7vm4,shadow-7vm6,shadow-7vm7,shadow-
7vm8,shadow-7vm9.shadow.whamcloud.com lctl set_param -n fail_loc=0          fail_val=0
2>/dev/null || true
done.
CMD: shadow-7vm4,shadow-7vm6,shadow-7vm7,shadow-7vm8 rc=\$([ -f /proc/sys/lnet/catastrophe ]
&&
      echo \$(< /proc/sys/lnet/catastrophe) || echo 0);
      if [ \$?rc -ne 0 ]; then echo \$(hostname): \$?rc; fi
      exit \$?rc

```

PASS 31c (4s)

```

== sanity-lfsck test 31d: Set broken striped directory (modified after broken) as read-only
== 06:35:35 (1409985335)

```

#####

For some reason, the master MDT-object of the striped directory may lost its master LMV EA. If somebody created files under the master directly after the master LMV EA lost, then the LFSCK should NOT re-generate the master LMV EA, instead, it should change the broken striped directory as read-only to prevent further damage

#####

```

rm: cannot remove `./mnt/lustre/.lustre/lost+found/MDT0001/[0x340000bd1:0xb:0x0]-P-0':
Directory not empty

```

Inject failure stub on MDT0 to simulate the case that the master MDT-object of the striped directory lost the LMV EA.

```

CMD: shadow-7vm8 /usr/sbin/lctl set_param fail_loc=0x1629
fail_loc=0x1629

```

```

CMD: shadow-7vm8 /usr/sbin/lctl set_param fail_loc=0x0
fail_loc=0x0

```

```

10.1.4.75@tcp:/lustre /mnt/lustre lustre rw,flock,user_xattr 0 0

```

```

CMD: shadow-7vm9.shadow.whamcloud.com grep -c /mnt/lustre' ' /proc/mounts
Stopping client shadow-7vm9.shadow.whamcloud.com /mnt/lustre (opts:)
CMD: shadow-7vm9.shadow.whamcloud.com lsof -t /mnt/lustre
CMD: shadow-7vm9.shadow.whamcloud.com umount /mnt/lustre 2>&1
Starting client: shadow-7vm9.shadow.whamcloud.com: -o user_xattr,flock shadow-7vm8@tcp:/lustre /mnt/lustre
CMD: shadow-7vm9.shadow.whamcloud.com mkdir -p /mnt/lustre
CMD: shadow-7vm9.shadow.whamcloud.com mount -t lustre -o user_xattr,flock shadow-7vm8@tcp:/lustre /mnt/lustre
Trigger namespace LFSCK to find out the inconsistency
CMD: shadow-7vm8 /usr/sbin/lctl lfsck_start -M lustre-MDT0000 -t namespace -r -A
Started LFSCK on the device lustre-MDT0000: scrub namespace
CMD: shadow-7vm8 /usr/sbin/lctl get_param -n mdd.lustre-MDT0000.lfsck_namespace |
    awk '/^status/ { print \$2 }'
CMD: shadow-7vm8 /usr/sbin/lctl get_param -n mdd.lustre-MDT0000.lfsck_namespace |
    awk '/^status/ { print \$2 }'
CMD: shadow-7vm8 /usr/sbin/lctl get_param -n mdd.lustre-MDT0000.lfsck_namespace
    File: `/mnt/lustre/d31d.sanity-lfsck/striped_dir/dummy'
    Size: 0          Blocks: 0          IO Block: 4194304 regular empty file
Device: 2c54f966h/743766374d      Inode: 216173000704131073  Links: 1
Access: (0644/-rw-r--r--)  Uid: (  0/   root)   Gid: (  0/   root)
Access: 2014-09-06 06:35:37.000000000 +0000
Modify: 2014-09-06 06:35:37.000000000 +0000
Change: 2014-09-06 06:35:37.000000000 +0000
touch: cannot touch `/mnt/lustre/d31d.sanity-lfsck/striped_dir/foo': Permission denied
Resetting fail_loc on all nodes...CMD: shadow-7vm4,shadow-7vm6,shadow-7vm7,shadow-7vm8,shadow-7vm9.shadow.whamcloud.com lctl set_param -n fail_loc=0 fail_val=0
2>/dev/null || true
done.
CMD: shadow-7vm4,shadow-7vm6,shadow-7vm7,shadow-7vm8 rc=\$([ -f /proc/sys/lnet/catastrophe ]
&&
    echo \$(< /proc/sys/lnet/catastrophe) || echo 0);
if [ \$rc -ne 0 ]; then echo \$(hostname): \$rc; fi
exit \$rc
PASS 31d (4s)

== sanity-lfsck test 31e: Re-generate the lost slave LMV EA for striped directory (1) ==
06:35:40 (1409985340)
#####
For some reason, the slave MDT-object of the striped directory
may lost its slave LMV EA. The LFSCK should re-generate the
slave LMV EA.
#####
rm: cannot remove `/mnt/lustre/.lustre/lost+found/MDT0001/[0x340000bd1:0xb:0x0]-P-0':
Directory not empty
Inject failure stub on MDT0 to simulate the case that the
slave MDT-object (that resides on the same MDT as the master
MDT-object resides on) lost the LMV EA.
CMD: shadow-7vm8 /usr/sbin/lctl set_param fail_loc=0x162a fail_val=0
fail_loc=0x162a

```

```

fail_val=0
CMD: shadow-7vm8 /usr/sbin/lctl set_param fail_loc=0x0 fail_val=0
fail_loc=0x0
fail_val=0
Trigger namespace LFSCK to re-generate slave LMV EA
CMD: shadow-7vm8 /usr/sbin/lctl lfsck_start -M lustre-MDT0000 -t namespace -r -A
Started LFSCK on the device lustre-MDT0000: scrub namespace
CMD: shadow-7vm8 /usr/sbin/lctl get_param -n          mdd.lustre-MDT0000.lfsck_namespace |
      awk '/^status/ { print \$2 }'
CMD: shadow-7vm8 /usr/sbin/lctl get_param -n          mdd.lustre-MDT0000.lfsck_namespace |
      awk '/^status/ { print \$2 }'
CMD: shadow-7vm8 /usr/sbin/lctl get_param -n mdd.lustre-MDT0000.lfsck_namespace
Resetting fail_loc on all nodes...CMD: shadow-7vm4,shadow-7vm6,shadow-7vm7,shadow-
7vm8,shadow-7vm9.shadow.whamcloud.com lctl set_param -n fail_loc=0          fail_val=0
2>/dev/null || true
done.
CMD: shadow-7vm4,shadow-7vm6,shadow-7vm7,shadow-7vm8 rc=\$( [ -f /proc/sys/lnet/catastrophe ]
&&
      echo \$(< /proc/sys/lnet/catastrophe) || echo 0);
      if [ \$rc -ne 0 ]; then echo \$(hostname): \$rc; fi
      exit \$rc
PASS 31e (3s)

```

```

== sanity-lfsck test 31f: Re-generate the lost slave LMV EA for striped directory (2) ==
06:35:43 (1409985343)

```

```
#####
```

For some reason, the slave MDT-object of the striped directory may lost its slave LMV EA. The LFSCK should re-generate the slave LMV EA.

```
#####
```

```
rm: cannot remove `mnt/lustre/.lustre/lost+found/MDT0001/[0x340000bd1:0xb:0x0]-P-0':
Directory not empty
```

Inject failure stub on MDT0 to simulate the case that the slave MDT-object (that resides on differnt MDT as the master MDT-object resides on) lost the LMV EA.

```
CMD: shadow-7vm8 /usr/sbin/lctl set_param fail_loc=0x162a fail_val=1
```

```
fail_loc=0x162a
```

```
fail_val=1
```

```
CMD: shadow-7vm8 /usr/sbin/lctl set_param fail_loc=0x0 fail_val=0
```

```
fail_loc=0x0
```

```
fail_val=0
```

Trigger namespace LFSCK to re-generate slave LMV EA

```
CMD: shadow-7vm8 /usr/sbin/lctl lfsck_start -M lustre-MDT0000 -t namespace -r -A
```

Started LFSCK on the device lustre-MDT0000: scrub namespace

```
CMD: shadow-7vm4 /usr/sbin/lctl get_param -n          mdd.lustre-MDT0001.lfsck_namespace |
      awk '/^status/ { print \$2 }'
```

```
CMD: shadow-7vm4 /usr/sbin/lctl get_param -n          mdd.lustre-MDT0001.lfsck_namespace |
      awk '/^status/ { print \$2 }'
```

```
CMD: shadow-7vm4 /usr/sbin/lctl get_param -n mdd.lustre-MDT0001.lfsck_namespace
```

```
Resetting fail_loc on all nodes...CMD: shadow-7vm4,shadow-7vm6,shadow-7vm7,shadow-7vm8,shadow-7vm9.shadow.whamcloud.com lctl set_param -n fail_loc=0 fail_val=0 2>/dev/null || true
```

done.

```
CMD: shadow-7vm4,shadow-7vm6,shadow-7vm7,shadow-7vm8 rc=\$([ -f /proc/sys/lnet/catastrophe ] &&
```

```
    echo \$(< /proc/sys/lnet/catastrophe) || echo 0);  
    if [ \${rc} -ne 0 ]; then echo \$(hostname): \${rc}; fi  
    exit \${rc}
```

PASS 31f (4s)