

Lustre 2.2 Test Plan

Revision History

Date	Revision	Author
2011-10-08	First draft	Peter Jones
2012-01-11	Second draft	Peter Jones
2012-01-23	Third draft	Peter Jones
2012-01-23	Add features for testing	Andreas Dilger
2012-01-27	Tidy up for latest status	Peter Jones
2012-02-09	Final details of feature testing	Peter Jones

2.2 Test Plan



2012-02-10	Descoping fc15 client support	Peter Jones
2012-03-14	Updating to use 2.1.1 instead of 2.1.0	Peter Jones

2.2 Test Plan



Table of Contents

[Release Goals](#)

[Changes from Previous Release](#)

[Support Matrices](#)

[Feature Test Areas](#)

[Wide Striping](#)

[Parallel Directory Operations/OSD API/mds-survey.sh](#)

[Imperative Recovery](#)

[Statahead/AGL](#)

[Functional Test Areas](#)

[Regression Testing](#)

[Stress Tests at Scale](#)

[Performance Testing](#)

[Interoperability](#)

[POSIX Compliance](#)

[Failover/Recovery Test](#)

[NFS](#)

[Multi-Mount Protection](#)

[Upgrade/Downgrade](#)

[Large LUN Testing](#)

2.2 Test Plan



Release Goals

The goal of this release is to provide a number of new Lustre features with quality that matches or surpasses Lustre 2.1.

Changes from Previous Release

The changes from Lustre 2.1 are:

- Deprecated RHEL\CentOS 5.x servers
- Deprecated i686 clients

Clients
Architectures
<ul style="list-style-type: none">• X86_64
Distributions & Kernels
<ul style="list-style-type: none">• RHEL 6.2• RHEL 5.7• SLES 11 SP1• TCP (1GigE)• IB (in-kernel OFED)• IB (external OFED 1.5.4)

Support Matrices

Servers
Architectures
<ul style="list-style-type: none">• X86_64
Distributions & Kernels
<ul style="list-style-type: none">• RHEL6.2

2.2 Test Plan

Feature Test Areas

For new features being added to the release, specific feature testing plans are defined below. The list of features being added to the 2.x release are:

- Wide Striping - up to 2000 stripes per file, up from 160 stripes per file
- Parallel Directory Operations - MDS parallel performance improvement
- mds_survey - local echo_client workload generator for MDD layer on MDS
- Imperative Recovery - MGS-driven server recovery speedup
- Statahead/AGL - `ls -l` performance improvement
- Multi-threaded ptrlpdc - client-side parallel RPC send/receive/checksum
- OSD transaction API - prepare OSD code for OFD and ZFS

Wide Striping

The Wide Striping feature will be tested manually/automated according to the test plan attached to the JIRA ticket - <http://jira.whamcloud.com/browse/LU-80>

Test Configuration
Regression tests 2.2 servers and clients
Lustre 1.8.7-wc1 clients and 2.2 servers
Lustre 2.1.1 clients and 2.2 servers

Parallel Directory Operations/OSD API/mds-survey.sh

Parallel Directory Operations will be tested manually/automated using the `mds-survey.sh` test script with the following configurations. This will serve the purpose of testing both PDO and OSD API features, and verifies that `mds-survey.sh` is running correctly.

NOTE: These tests should now be fully automated

Test Configuration
RHEL6 server

Imperative Recovery

Imperative Recovery will be tested manually/automated using the Imperative Recovery test plan with the following configurations. The test plan is attached to the JIRA ticket - <http://jira.whamcloud.com/browse/LU-19>

2.2 Test Plan

NOTE: These tests should now be fully automated

Test Configuration

RHEL6 server and client

Statahead/AGL

Statahead and Asynchronous Glimpse Lock (AGL) will be automated by updating the mdsrate-stat-large.sh test script to include a phase which measures the single-client `ls -l` performance on a 1M and 10M entry directory and compare it with test results using a Lustre 1.8.7 and 2.1.1 client with the following configurations. <http://jira.whamcloud.com/browse/LUDOC-42>

Test Configuration

RHEL 6.2 servers - RHEL 6 client

RHEL 6.2 servers - RHEL 5 client

RHEL 6.2 servers - SLES11 client

Functional Test Areas

The below functional test areas are all run manually except the Regression testing section.

Regression Testing

Use [auster](#) to run automated regression tests with the following configurations. <http://wiki.whamcloud.com/display/ENG/Auster> .

Test Configuration

RHEL6 Servers – RHEL 6 Clients

Inkernel OFED – X86_64

RHEL6 Servers – RHEL 5 Clients

Inkernel OFED – X86_64

RHEL6 Servers – SLES 11 Clients

Inkernel OFED – X86_64

RHEL6 Servers – RHEL 6 Clients

External OFED – X86_64

2.2 Test Plan

Stress Tests at Scale

Execute Hyperion-sanity on Hyperion cluster

<http://wiki.whamcloud.com/display/ENG/Hyperion+Tests>

Test Configuration
run parallel-scale auster test with maximum number of clients. Subtests include <ul style="list-style-type: none">- ior(shared/fpp), mdtest, simul, mdsrate
run recovery-scale auster test for 24 hours each flavor (MDS/OSS) with maximum number of clients

Performance Testing

The performance test plan should be updated with each release to take into account any new features that may have impacts on Lustre performance, and should note landings/bugfixes that may impact performance, or require performance validation.

The basic performance testing will comprise bulk data transfer, file creation and network tests using both single-shared file and file-per-process methods where applicable. The current test plan will use IOR(POSIX), mdsrate and lnet_self with other tests to be added as needed.

Testing will use a constant number of clients for each release to facilitate run-to-run comparison. Tests will be run with 100 and 50 clients, given current Hyperion scale and for comparison with previous performance test results.

Results will be compared to the previous release of Lustre and the "bare metal" baseline (obtained from odbfilter-survey and lnet-selftest) on the same test configuration.

Results less than the previous version will be investigated, results within 5% may be considered within normal variation, subject to investigation. Runs resulting in issues and/or performance degradation greater than 5% will be marked as failed. Runs showing performance improvement greater than 10% will be checked for rationality issues such as improper test parameters.

Test Configuration
Performance <ul style="list-style-type: none">- run IOR with 50 and 100 clients (shared and fpp)- run mdsrate with 50 and 100 clients- run lnet_selftest with 50 and 100 clients

2.2 Test Plan

Interoperability

Interoperability testing between both 1.8.7-wc1 and 2.1.1 clients with 2.2 servers. Also, 2.1.1 server with 2.2 clients. This is supported on our autotest system on Toro and not run manually <http://wiki.whamcloud.com/display/ENG/Auster>

Test Configuration
Quotas- RHEL6 1.8.7 client-- <input type="checkbox"/> RHEL6 2.2 server
Quotas- RHEL6 2.1.1 client-- <input type="checkbox"/> RHEL6 2.2 server
Quotas- RHEL6 2.1.1 server RHEL6 2.2 client

POSIX Compliance

Execute tests to ensure POSIX compliance.

<http://wiki.whamcloud.com/display/ENG/POSIX+Compliance+Testing>

Test Configuration
POSIX test RHEL6 client

Failover/Recovery Test

Execute recovery and failover testing for hard failure mode (powering off and on) with shared storage in server failover pairs, manually. Soft failover is covered by the auster Regression test suite.

<http://wiki.whamcloud.com/display/ENG/Failover%2C+Recovery%2C+and+MMP>

NOTE: These tests should now be fully automated

Test Configuration
Recovery test RHEL6 client
Recovery test RHEL5 client
Recovery test SLES11 client

NFS

Execute NFS v3/v4 testing using the following configurations.

<http://wiki.whamcloud.com/display/ENG/NFSv3+and+v4+Over+Lustre>

NOTE: These tests should now be fully automated

2.2 Test Plan

Test Configuration
Parallel-scale RHEL6 client
Parallel-scale RHEL5 client
Parallel-scale SLES11 client

Multi-Mount Protection

This test is included in the auster suite but is run manually (powering off and on) and server failover pairs with shared storage to verify the Multiple Mount Protection functionality.

<http://wiki.whamcloud.com/display/ENG/Failover%2C+Recovery%2C+and+MMP>

NOTE: These tests should now be fully automated

Test Configuration
RHEL6 client

Upgrade/Downgrade

Execute clean and rolling upgrade and downgrade testing from both 1.8.7-wc1 and 2.1 to 2.2.

<http://wiki.whamcloud.com/display/ENG/Upgrade+and+Downgrade+Testing>

Test Configuration
OSS1: upgrade from 1.8.7-wc1(RHEL5/x86_64) to 2.1.1(RHEL5/x86_64), then upgrade to 2.2(RHEL6/x86_64), then downgrade to 2.1.1(RHEL5/x86_64)
OSS2: upgrade from 2.1.1(RHEL6/x86_64) to 2.2(RHEL6/x86_64), then downgrade to 2.1.1(RHEL6/x86_64)
MDS: upgrade from 1.8.7-wc1(RHEL5/x86_64) to 2.1.1(RHEL6/x86_64), then upgrade to 2.2(RHEL6/x86_64), then downgrade to 2.1.1(RHEL6/x86_64)
Client1: upgrade from 1.8.7-wc1(RHEL6/x86_64) to 2.1.1(RHEL6/x86_64), then upgrade to 2.2(RHEL6/x86_64), then downgrade to 2.1.1(RHEL6/x86_64)
Client2: upgrade from 1.8.7-wc1(RHEL5/x86_64) to 2.1.1(RHEL5/x86_64), then upgrade to 2.2(RHEL5/x86_64), then downgrade to 2.1.1(RHEL5/x86_64)

2.2 Test Plan

Large LUN Testing

Execute large LUN feature testing.

<http://wiki.whamcloud.com/display/ENG/Large+LUN+Testing>

Test Configuration
Juelich