Lustre Test Plan

Large Scale Test Plan

Clustered MetaData

(only for non-production system)

Author	Date	Description of Document Change	Client Approval By	Client Approval Date
Elena Gryaznova	2008-10-21	draft		

I. Test Plan Overview

Executive Summary

- Statement of the problem trying to solve:
  Test at scale Clustered MetaData (CMD) feature of 2.0

- Required inputs:
  -- This test plan approved by customer
  -- The test scripts reviewed by customer and adjusted by QE to use in customer testing lab
  -- Estimated debug impact for required debug level
  -- The packages:
  Lustre HEAD build, lustre-tests HEAD build.
- Hardware to be used:
  -- Customer lab
- Expected output:
  -- The current CMD status of HEAD (2.0 release?)
  -- The following tickets will be used for summary and a status of testing:
  bug 17737 “Large Scale HEAD CMD testing”
  bug 17345 “HEAD CMD acc-sm testing status”

Problem Statement

QE is required to testing of CMD feature at scale. This test plan describes the tests need to be run to verify this feature.

Goal

Verify MD clustering functionality on a large system.
Get the performance comparison CMD vs. single MDT configuration.

Success Factors

All tests need to run successfully.

Performance improvements in comparison with a single MDT configuration.

Testing Plan

Define the setup steps that need to happen for the hardware to be ready? Who is responsible for these tests?
1. Get system time at customer lab. 2. Install Lustre builds. 3. Configure Lustre file system. 4. Pre-feature testing has been completed (including acceptance-small for CMD configuration, MDSCOUNT=4).

QE team members responsible for :
Elena Gryaznova – creating the tests scripts, adjust scripts for customer test environment.
Elena Gryaznova – debug level impact estimation
– review the tests scripts, help to adjust them to customer test environment.
– setting up the test environment, running the tests, reporting the results, collect debug data of failed tests.

Developers responsible for:
Yury Umanets – review the test plan, inspect the tests scripts. Figure out and resolve the failures.

Specify the date these tests will start, and length of time that these test will take to complete.

Date started: TBD

The time estimation for new test creation by QE: 5 days

The time estimation of 1 run:
Performance mdsrate tests: 2 days
MDS scaling test: 2 days
Common recovery tests: 4 days

Summary for 1 tests cycle: 8 days

In the case of defects found the tests should be repeated.
The estimated number of cycles: TBD

The estimated number of cycles and completed testing depends on:
-- the number of defects found during testing;
-- the time needed by developers to fix the defects;

Specify (at a high level) what tests will be completed?

Functional tests: new acceptance-small tests: large-scale, recovery-scale (with Lustre file system reformatting with CMD).

Specify how you will restore the hardware (if required) and notify the client your testing is done.

QE will need feedback from customers, use bug 17737 for outputs.

The bugzilla ticket is filed for each failure, use bug 17737 as blocker bug.

The following debug data need to be collected for failed tests and attached to bug tickets:
/with default acc-sm ptl and subsystem debug level/*
1) Lustre debug logs from MDS/OST nodes
2) Lustre debug logs from all clients (if possible)
3) syslog from MDS/OST nodes
4) syslog from all client nodes (if possible)

* Confirmed by Yury Umanets.

Summary and status reports are printed in the bug 17737 “large scale HEAD CMD testing”

Test Cases

Requirements:
1. Installed Lustre user tools (lctl, lfs)
2. Shared directory with lustre-tests build on all clients
3. Installed mdsrate, mpirun, IOR, dbench, iozone

I. performance mdsrate tests

All these tests are integrated into acceptance-small as performance-sanity.sh (PERFORMANCE_SANITY).
To run this performance-sanity test:
1. Install Lustre build and lustre-tests build on lab
2. Specify the cluster configuration file, see cfg/local.sh and cfg/ncli.sh for details. Use MDSCOUNT=4.
3. Run the test with lustre reformatting as:

```
    REFORMAT=--reformat ACC_SM_ONLY=PERFORMANCE_SANITY NAME=<config_file> sh acceptance-small.sh
or
    SETUP=: CLEANUP=: FORMAT=: NAME=<config_file> sh large-scale.sh
```

performance mdsrate tests includes:
- small file creation performance test: create/open/delete
- large file creation performance test: create/open/delete
- lookup rate 10M file 1 dir
- lookup rate 10M file 10 dirs
- file attribute retrieval rate for small file creation
- file attribute retrieval rate for large file creation

1.a performance-sanity for configuration (NC:1M:MO)
1.b performance-sanity for configuration (NC:4M:MO)
1.c performance-sanity for configuration (NC:16M:MO)

II. MDS scaling test

2.a (was test46 in cmd3 project) (NC:16M:MO)
Create/unlink/lookup/stat mdsrate operations on a configurations with number of MDSs vary from 1 to maximum (16) with the appropriate number of clients and threads. Use only the configuration where the number of OSTs is greater than or equal to the number of MDSs. These results will show how the workload is shared by the MDSs present.

This test is (will be) integrated into acceptance-small as mds-scale-load.sh (MDS_SCALE_LOAD).

To run this scale tests over acceptance-small:

```
    REFORMAT=--reformat ACC_SM_ONLY=MDS_SCALE_LOAD NAME=<config_file> sh acceptance-small.sh
```

```
To run this test separately:
    REFORMAT=--reformat NAME=<config_file> mds-scale-load.sh
```

COMMON scale tests

All these tests are (will be) integrated into acceptance-small as recovery-scale.sh (RECOVERY_SCALE).

```
To run this recovery scale tests over acceptance-small:
    SETUP=: CLEANUP=: FORMAT=: ACC_SM_ONLY=RECOVERY_SCALE
NAME=<config_file> sh acceptance-small.sh
To run all recovery scale tests:
    SETUP=: CLEANUP=: FORMAT=: NAME=<config_file> sh recovery-scale.sh
To run the recovery scale tests separately:
    SETUP=: CLEANUP=: FORMAT=: NAME=<config_file> DURATION=<duration>
recovery-mds-scale.sh
    SETUP=: CLEANUP=: FORMAT=: NAME=<config_file> recovery-double-scale.sh
```

III.
Scale recovery tests (NC:16M:MO) (the test based on test11/17 from CMD3 project)

Scale recovery tests (NC:16M:MO) (the test based on test11/17 from CMD3 project)

3.a (was test11 in cmd3 project)
For a defined duration (1-24 hours), repeatedly fail an random MDS at defined (5-10 minutes) intervals and verify that no application errors occur. Load of clients: mdsrate, IOR, dbench, iozone.

recovery-mds-scale.sh

Example

```
    SETUP=: CLEANUP=: REFORMAT=--reformat: NAME=<config_file>
DURATION=”3600” sh recovery-mds-scale.sh
```

3.b (was test17 on cmd3 project)
Fail a random pair of nodes at defined (5-10 minutes) intervals and verify that no application errors occur. Load of clients: mdsrate,IOR, dbench, iozone.

1: failover MDS, then OST
2: failover MDS, then 2 clients
3: failover MDS, then another MDS
4: failover OST, then another OST
5: failover OST, then 2 clients
6: failover OST, then MDS
7: failover random 2 clients, then MDS
8: failover random 2 clients, then OST
9: failover random 2 clients, then 2 different clients

recovery-double-scale.sh

Example:

```
  SETUP=: CLEANUP=: FORMAT=: NAME=<config_file> sh recovery-double-
scale.sh
```

IV. More tests can be added after discussion.

Benchmarking

No additional benchmarks will be done.

II. Test Plan Approval

Review date for the Test Plan review with the client:

- 11/21/08 – send to review by Yury Umanets
-
- Date(s) agreed to by the client to conduct testing

III. Test Plan – Final Report

Test Results

Benchmarking

Conclusions

Summary of the test:

-
-

Next Steps

-
-
-