Lustre Test Plan
Version recovery phase 2
Large Scale Test Plan

Author	Date	Description of Document Change	Client Approval By	Client Approval Date
Elena Gryaznova	2008-10-17	draft		
Elena Gryaznova	2008-11-20	correct in according to Nicholas Henke review		
J.D. Neumann	01/29/09	Additional changes and corrections		

I. Test Plan Overview

Executive Summary

- Statement of the problem trying to solve:
  Test at scale the following version based recovery feature landed into b1_8_gate branch
  -- vbr_interop
  -- vbr_exp
  -- vbr_orphans
- 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 (have to be <5%)
  -- b1_8_gate branch, with landed :
  vbr_interop: attachment 17435, attachment 17694 bug 15942
  vbr_exp: attachment 18379 bug 15391
  vbr_orphans: attachment (still in code phase) bug 15292
  -- the packages (build): Lustre b1_8_gate build, lustre-tests b1_8_gate build.
- Hardware to be used:
  -- customer lab
- Expected output:
  -- The current status of the listed features
  -- The following tickets will be used for summary and a status of testing.
  vbr_interop: bug 15942
  vbr_exp: bug 15391
  vbr_orphans: bug 15392
  large-scale VBR tests: bug 17195 “create VBR large scale tests”
  large-scale VBR testing status: **bug 17739** “large scale VBR testing”

Problem Statement

We need to test VBR at scale. This test plan lays out what tests need to be run to verify this feature.

Goal

Verify the VBR functionality on a large system.
Verify VBR performance impact on a large system.

Success Factors

All tests need to run successfully.

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 rpms. 3. Configure Lustre file system. 4. pre-feature testing has been completed (including replay-vbr on 2 clients) and signed off by SUN QE for VBR feature.

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

Developers responsible for:
Mikhail Pershin – 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 tests creation: 1 week The time estimation of 1 run: VBR feature tests (large-scale) : 2 days Common recovery tests : 4 days Summary for 1 tests cycle: 6 days In the case of defects found the tests should be repeated.   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 (without Lustre filesystem reformatting)

Test Cases

Test Cases
Post-gate landing
All these tests are (will be) integrated into acceptance-small as large-scale.sh (LARGE_SCALE). To run this large scale 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. 3. run the test without lustre reformatting as: SETUP=: CLEANUP=: FORMAT=: ACC_SM_ONLY=LARGE_SCALE NAME=<config_file> sh acceptance-small.sh or

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

Requirements:
1. installed Lustre user tools (lctl)
3. shared directory with lustre-tests build on all clients
4. formatted Lustre fs, mounted by clients
5. the configuration file in according to formatted lustre
6. installed mdsrate, IOR, mpirun, dbench, iozone
```

I.

Feature tests for exports:

```
1.a measure N clients connection time without delayed exports (and
orphans).

1.b create many delayed exports 1000/10000/100000,  measure time for
clients connection time - connect N new clients
large_scale.sh test_1b

1.c  create many delayed exports 1000/10000/100000, measure recovery time:
connect several clients, create/delete the number of files, fail mds.
Print the statistic number of delayed exports/ the number of clients/ the
time of connections
large_scale.sh test_1c

1.d   create many delayed exports 1000/10000/100000,  expire all exports
and connect new clients, this will invoke massive orphans cleanup. Measure
connection time.
large_scale.sh test_1d

 1.e create many delayed exports 1000/10000/100000, make export expired
one-by-one with  delay 30 sec, measure recovery time: connect several
clients, create/delete the number of files, fail mds. Print the statistic
number of delayed exports/ the number of clients/ the time of connections.
This should invoke contstant exports+orphans cleanup during test.
large-scale.sh test_1e
```

II.

Feature tests for orphans

```
2.b  create many delayed exports 1000/10000/100000 with orphaned files
(1000/10000/100000), measure time for clients connection time - connect N
new clients
large_scale.sh test_2b

2.c create many delayed exports 1000/10000/100000 with orphaned files
(1000/10000/100000),  measure recovery time: connect several  clients,
create/delete the number of files, fail mds. Print the statistic number of
delayed exports/ the number of clients/ the time of connections
large_scale.sh test_2c

2.d create many delayed exports 1000/10000/100000 with orphaned files
(1000/10000/100000), measure the connection time.
large_scale.sh test_2d
```

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:1M:MO) (the test based on test11/17 from CMD3 project)

```
Scale recovery tests (NC:1M:MO) (the test based on test11/17 from CMD3
project)

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

recovery-mds-scale.sh

Example:
    SETUP=: CLEANUP=: FORMAT=: 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, dd, tar.
1: failover MDS, then OST
2: failover MDS, then 2 clients
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
```

```
3.c client failure does not affect other clients
Start load on clients. At defined (5-10 minutes) interval fail one random
client and then fail mds. Reintegrate failed client after recovery
completed. Load of clients: mdsrate,IOR, dbench, iozone.
```

```
recovery-random-scale.sh

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

IV. More tests can be added after discussion (see bug 17195).

Benchmarking

No additional benchmarks will be done.

I. Test Plan Approval

Review date for the Test Plan review with the client:

- 10/16/08 – reviewed by Mikhail Pershin
- 10/17/08 – reviewed by J.D. Neumann
- 10/18/08 – reviewed by Ed Giesen
- 11/11/08 – reviewed by Nicholas Henke
- 11/20/08 – reviewed by Mikhail
- Date the Test Plan was approved by the client (and by whom)
  08/11/2008 – approved by Charlie Carroll

- Date(s) agreed to by the client to conduct testing

I. Test Plan – Final Report

Test Results