Lustre OSS Read Cache Feature SCALE Test Plan

Author	Date	Description of Document Change	Client Approval By	Client Approval Date
Jack Chen	10/08//08	Add scale testing to Feature Test Plan		
Jack Chen	10//30/08	Create Scale Test Plan, draft		
Jack Chen	11/04/08	Add performance with OSS read cache enabled/disabled		
Jack Chen	11/06/08	Add test goal by Alex's suggestion		

I. Test Plan Overview

This test plan describes various testing activities and responsibilities that are planned to be performed by Lustre QE. Per this test plan, Lustre QE will provide large-scale, system level testing for the OSS read cache feature.

Executive Summary

- Create a test plan to provide testing for the OSS read cache project for a large-scale cluster.
- Required input from developers.
- Require customer large cluster lab.
- The output will be all tests are passed.

Problem Statement

We need to test the OSS read cache feature on a large-scale cluster to make sure the feature is scalable.

Goal

Verify that OSS read cache functions with a large system.
Compare performance data between OSS read cache enabled/disabled.

1) See improvement in those specific cases.
2) Prove there is no noticeable regression in the other cases.

Success Factors

All tests need to run successfully.

Performance data of OSS read cache enabled should be better then OSS read cache disabled.

Testing Plan

Pre-gate landing

Define the setup steps that need to happen for the hardware to be ready? Who is responsible for these tests?
Get system time on a large system. Pre-feature testing has been completed and signed off by SUN QE for this feature.

Specify the date these tests will start, and length of time that these test will take to complete.
Date started: 2008-11-05
The time estimation for new test creation: 1 week The time estimation of 1 run: OSS read cache feature tests (large-scale) : 2 ~ 3 days Performance tests: 2 ~ 3 days

* In the case of defects found, the tests should be repeated. The estimated time to complete testing depends on: -- the number of defects found during testing; -- the time needed by a developer to fix the defects;

Specify (at a high level) what tests will be completed? New, Exist tests, manual or automate	
IOR, PIOS	Existing test, automate

Specify how you will restore the hardware (if required) and notify the client your testing is done.
We will need feedback from the user; recommend that we use bugzilla for outputs.
The bugzilla ticket is filed for each failure.
Summary and status report are printed in the bug that we create for this test.

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.rpm and lustre-tests.rpm on all cluster nodes. 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 build packages on all cluster nodes. 2. Installed Lustre user tools (lctl). 3. Shared directory with lustre-tests build on all clients 4. Formatted Lustre file system, mounted by clients 5. The configuration file in according to a formatted Lustre system 6. Installed dd, tar, dbench, iozone

Performance compare test data between cache enabled and disabled.		
No.	Test Case	Description
1. IOR	Run IOR with OSS cache disabled	Create a file system with all of the OSTs and set stripe across all. Run IOR using mpirun from one client and record the performance numbers.

		Install and configure Lustre first. Manual steps to run IOR: 1) disable OSS cache lctl set_param -n odbfilter.*.cache 0 or lctl set_param -n obdfilter.*.read_cache_enable 0 lctl set_param -n obdfilter.*.writethrough_cache_enable 0 2) disable debug /sbin/sysctl -w lnet.subsystem_debug=0 /sbin/sysctl -w lnet.debug=0 3) set stripe /usr/bin/lfs setstripe ${MOUNTPOINT} 1048576 -1 -1 4) run IOR as mpiuser ${MPIRUN} -machinefile ${MACHINE_FILE} -np ${NPROC} ${IOR} -t $bufsize -b $blocksize -f $input_file Note: MACHINE_FILE list all Lustre clients we used NPROC is the number of MPI processes, the better number is the client number BUFSIZE is 1M BLOCKSIZE depends on how many memory blocks are on all Lustre client nodes, $Total_client_memory X 2 = $BLOCKSIZE x $NPROC 5) collect “vmstat 1” and “oprofiles” for all the runs - this is very important data.
	Run IOR with OSS cache enabled	The different step is 1) enable OSS read cache first. 1) enable OSS cache lctl set_param -n odbfilter.*.cache 1 or lctl set_param -n obdfilter.*.read_cache_enable 1 lctl set_param -n obdfilter.*.writethrough_cache_enable 1
2. PIOS	Run PIOS with OSS read cache disabled	Create a file system with all of the OSTs and set stripe across all. Run PIOS using pdsh from all of the clients and record the performance numbers. Install and configure Lustre first. Manual steps to run PIOS: 1) disable OSS cache lctl set_param -n odbfilter.*.cache 0 or lctl set_param -n obdfilter.*.read_cache_enable 0 lctl set_param -n obdfilter.*.writethrough_cache_enable 0 2) disable debug /sbin/sysctl -w lnet.subsystem_debug=0 /sbin/sysctl -w lnet.debug=0 3) set stripe /usr/bin/lfs setstripe ${MOUNTPOINT} 1048576 -1 -1

		4) run PIOS as mpiuser $PIOS -t 1,8,16,40 -n 1024 -c 1M -s 8M -o 16M -p $LUSTRE $PIOS -t 1,8,16,40 -n 1024 -c 1M -s 8M -o 16M -p $LUSTRE verify $PIOS -t 1,8,16,40 -n 1024 -c 1M -s 8M -o 16M -L fpp -p $LUSTRE $PIOS -t 1,8,16,40 -n 1024 -c 1M -s 8M -o 16M -L fpp -p $LUSTRE verify 5) collect “vmstat 1” and “oprofiles” for all the runs - this is very important data
	Run PIOS with OSS read cache enabled	The different step is 1) enable OSS read cache first. 1) enable OSS cache lctl set_param -n odbfilter.*.cache 1 or lctl set_param -n obdfilter.*.read_cache_enable 1 lctl set_param -n obdfilter.*.writethrough_cache_enable 1

Benchmarking

No benchmarks will be done.

II. Test Plan Approval

- Review date for the Test Plan review with the client:
- Date the Test Plan was approved by the client (and by whom)
- 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

-
-