

Auster

STATUS:

This page is no longer maintained. Please see [Test Descriptions](#) for current test descriptions.

This page provides quick description on auster tests.

runtests

Simple basic regression test that verifies data persistence across write, unmount, and remount. This is one of the few tests that verifies data integrity across a full filesystem shutdown and remount, unlike many tests which at most only verify the existence/size of files.

sanity

A set of regression tests that verify operation under normal operating conditions. This tests a large number of unusual operations that have previously caused functional or data correctness issues with Lustre. Some of the tests are Lustre specific, and hook into the Lustre fault injection framework using the "lctl set_param fail_loc=X" command to activate triggers in the code to simulate unusual operating conditions that would otherwise be difficult or impossible to simulate.

fsx

The file system exerciser was developed outside of Lustre and is intended to stress test unusual IO and file operations. It does buffered and mmap reads and writes with random size and offset, truncate, open, close on a file. It verifies data integrity after each step. A lustre-developed extension allows fsx to run against multiple mountpoints to verify distributed IO operation correctness.

sanityn

Tests that verify operations from two clients under normal operating conditions. This is done by mounting the same filesystem twice on a single client, in order to allow a single script/program to execute and verify filesystem operations on multiple "clients" without having to be a distributed program itself.

lfsck

Tests e2fsck and lfsck to detect and fix filesystem corruption

liblustre

Runs a test linked to a liblustre client library

replay-single

A set of unit tests that verify recovery after MDS failure

conf-sanity

A set of unit tests that verify the configuration tools, and runs Lustre with multiple different setups to ensure correct operation in unusual system configurations.

recovery-small

A set of unit tests that verify RPC replay after communications failure

replay-ost-single

A set of unit tests that verify recovery after OST failure

replay-dual

A set of unit tests that verify the recovery from two clients after server failure

insanity

A set of tests that verify the multiple concurrent failure conditions

sanity-quota

A set of tests that verify filesystem quotas

The acceptance-small.sh is a wrapper which is normally used to run all (or any) of these scripts. In addition it is used to run the following pre-installed benchmarks:

dbench

Dbench benchmark for simulating N clients to produce the filesystem load

bonnie

Bonnie++ benchmark for creation, reading, and deleting many small files

iozone

iozone benchmark for generating and measuring a variety of file operations.