



# Using Lustre Test Framework and Parameters

Yu, Jian  
Feb. 27, 2013



# Agenda

- Lustre Test Suite Overview
- Lustre Test Framework
  - Test Framework Environment Variables
  - Common Test Framework Functions
- Autotest Test Parameters

# Lustre Test Suite Overview

Located in `lustre/tests/` directory of Lustre source code tree and `lustre-{client-,}tests-*.rpm`

- `llmount.sh` and `llmountcleanup.sh`
  - format, set up and clean up a Lustre filesystem
- `auster`
  - a test suite which wraps Lustre regression test suites (sanity, replay-single, parallel-scale, etc.)
  - a script to run Lustre regression test suites
  - usage: `auster -h`

# Test Framework Environment Variables

- default values are defined in `lustre/tests/cfg/{local,ncli}.sh` and `init_test_env()`
- non-default values can be defined and used as follows:
  - defined in a custom test configuration file `cfg/name.sh` which sources `cfg/ncli.sh` and passed to `auster` via `"-f name"` option or `"NAME=name"` variable
    - e.g., `auster -v -r -s -k -f my_config -g test-groups/full`
    - e.g., `NAME=my_config bash auster -v -r -s -k -g test-groups/full`
  - specified with variables and passed to `auster` directly
    - e.g., `PDSH="pdsh -t 120 -S -w" NAME=ncli RCLIENTS="client2 client3" mds_HOST=mds ost_HOST=oss bash auster -d /test_logs -v -r -s -k sanity --only 71`

# Test Framework Environment Variables (contd.)

- multiple remote clients
  - NAME=ncli RCLIENTS="client2 client3 ..."
    - "RCLIENTS" supports pdsh HOSTLIST expressions
      - e.g., RCLIENTS=client[2-10]
      - do not need specify "CLIENTS" variable which will be defined in `init_clients_lists()`
- backend filesystem type (ldiskfs, zfs)
  - FSTYPE
  - MGSFSTYPE, MDSFSTYPE, OSTFSTYPE
  - mgs\_FSTYPE, mds\${n}\_FSTYPE, ost\${n}\_FSTYPE

# Test Framework Environment Variables (contd.)

- debugging Lustre test scripts and Lustre codes
  - VERBOSE=true
    - output the command performed on remote node
    - output the mkfs.lustre parameters and progress
  - PTLDEBUG
    - `lctl set_param debug="$PTLDEBUG"` on all nodes
  - SUBSYSTEM
    - `lctl set_param subsystem_debug="$SUBSYSTEM"` on all nodes
  - DEBUG\_SIZE
    - `lctl set_param debug_mb="$DEBUG_SIZE"` on all nodes
  - DAEMONFILE and DAEMONSIZE
    - `lctl debug_daemon start $DAEMONFILE $DAEMONSIZE`

# Test Framework Environment Variables (contd.)

- failover testing with shared storage
  - mds\${n}failover\_HOST, ost\${n}failover\_HOST
    - failover partner node for mds\$n or ost\$n
  - MGSNID
    - for combined MGS/MDS, both the NIDs (separated by colon) of the MGS/MDS failover pair should be specified
  - FAILURE\_MODE
    - "SOFT": unmount and mount
    - "HARD": power off (without unmounting) and on
  - POWER\_DOWN, POWER\_UP
    - used in "HARD" failure mode to power off and on the remote test node

# Common Test Framework Functions

- defined in `lustre/tests/{test-framework,functions}.sh`
  - initiate test environment variables
  - perform command/function on remote nodes
  - format/set up/clean up Lustre filesystem
  - set debug level/size
  - error handling
  - AT/quotas/OST pools, etc. feature related
  - failover/recovery related
  - `Inet_selftest` related
  - MPI programs related, and so on.
- used for creating Lustre regression test suites/cases



# Common Test Framework Functions (contd.)

- `init_test_env()`
  - test environment variables: `PATH`, `TMP`, `TESTSUITE`, `LOGDIR`, `HOSTNAME`, `NAME`, etc.
  - e2fsprogs utils: `MKE2FS`, `DEBUGFS`, `TUNE2FS`, `E2FSCK`, `LFSCK_BIN`, etc.
  - Lustre utils: `LCTL`, `LFS`, `MKFS`, `TUNEFS`, etc.
  - ZFS utils: `ZFS`, `ZPOOL`, `ZDB`
  - needs to be improved to clean up test configuration files (LU-862)

# Common Test Framework Functions (contd.)

- `do_node()`, `do_nodes()`, `do_facet()`
  - perform command on remote node(s)
    - e.g., `do_facet $SINGLEMDS $LCTL set_param fail_loc=0`
    - e.g., `do_nodes $(comma_list $(nodes_list)) $LCTL clear`
- `do_rpc_nodes()`
  - perform common function on remote node(s)
    - e.g., `do_rpc_nodes $(facet_active_host $SINGLEMDS) is_mounted $MOUNT`
    - e.g., `do_rpc_nodes $(comma_list $(remote_nodes_list)) load_modules_local`

# Common Test Framework Functions (contd.)

- `facet_active_host()`
  - get the active host for the `$facet` in a failover test environment
  - use `$(facet_active_host $facet)` to get the host of `$facet`
- `mgsdevname()`, `mdsdevname()`, `ostdevname()`
  - use function instead of `MGSDEV`, `MDSDEV$n`, `OSTDEV$n`
- `facet_fstype()`
  - get the backend filesystem type of `$facet` in the order of `<facet>_FSTYPE`, `<facet_type>FSTYPE`, `FSTYPE`
  - use `facet_fstype()` instead of checking variables directly
- `lfs_df()`
  - use `lfs_df()` instead of `"lfs df"` or `"$LFS df"`

# Common Test Framework Functions (contd.)

- `version_code()`, `lustre_version_code()`
  - check Lustre version in the feature-related tests to make sure the tests do not run on unsupported Lustre version
    - e.g., 

```
[[ $(lustre_version_code $SINGLEEMDS) -ge $(version_code 2.3.59) ]] ||  
    { skip "Need MDS version at least 2.3.59"; return 0; }
```
- `generate_logname()`
  - used in sub-test to generate Maloo upload-able log file name
    - e.g., 

```
changelog_file=$(generate_logname "changelog")  
$LFS changelog $MDT0 > $changelog_file
```

# Common Test Framework Functions (contd.)

- `mdts_nodes()`, `osts_nodes()`, `remote_nodes_list()`, `nodes_list()`
  - single MDS facet: `$SINGLEMDS`
  - local Client node: `$HOSTNAME`
  - list of remote Client nodes: `$(comma_list $RCLIENTS)`
  - list of all Client nodes: `$CLIENTS`
  - list of all MDS nodes: `$(comma_list $(mdts_nodes))`
  - list of all OSS nodes: `$(comma_list $(osts_nodes))`
  - list of all Server nodes: `$(comma_list $(mdts_nodes) $(osts_nodes))`
  - list of all remote nodes: `$(comma_list $(remote_nodes_list))`
  - list of all nodes: `$(comma_list $(nodes_list))`
- `comma_list()`, `expand_list()`, `exclude_items_from_list()`

# Common Test Framework Functions (contd.)

- `start_full_debug_logging()`, `stop_full_debug_logging()`
- `save_lustre_params()`, `restore_lustre_params()`
- `set_nodes_failloc()`, `clear_failloc()`, `reset_fail_loc()`
- `wait_update()`, `wait_update_facet()`
- `wait_recovery_complete()`, `wait_mds_ost_sync()`
- `wait_import_state()`, `wait_osc_import_state()`,  
`wait_clients_import_state()`
- `sync_all_data()`, `wait_delete_completed_mds()`,  
`wait_destroy_complete ()`, `wait_delete_completed()`
- .....

**Please use, improve and create common functions.**

# Autotest Test Parameters

- add "Test-Parameters:" into commit messages to influence the testing performed on the patch
- tests specified via Test-Parameters are in addition to the normal "review" test group that would be run on the patch
- "fortestonly" parameter means the patch is not intended for landing, only the tests specified via Test-Parameters would be run
- **while creating patches, always check whether a regression test case/suite is needed or not**
- **while writing commit messages, always check "Test-Parameters: envdefinitions=SLOW=yes testlist=xxx" is needed or not**

# Autotest Test Parameters (contd.)

- ZFS testing

Test-Parameters: envdefinitions=SLOW=yes \  
mdtfilesystemtype=zfs mdsfilesystemtype=zfs ostfilesystemtype=zfs \  
testlist=sanity

- OFA testing

Test-Parameters: fortestonly \  
envdefinitions=SLOW=yes,ENABLE\_QUOTA=yes \  
clientdistro=el6 serverdistro=el6 \  
clientjob=lustre-master clientbuildno=1270 \  
serverjob=lustre-master serverbuildno=1270 \  
nettypes=o2ib clientibstack=ofa serveribstack=ofa testgroup=full



# Autotest Test Parameters (contd.)

- interoperability testing

```
Test-Parameters: envdefinitions=SLOW=yes \  
serverjob=lustre-b2_1 serverbuildno=164 \  
testlist=conf-sanity
```

```
Test-Parameters: fortestonly \  
envdefinitions=SLOW=yes,ENABLE_QUOTA=yes \  
clientdistro=el6 serverdistro=el6 \  
clientarch=x86_64 serverarch=x86_64 \  
clientjob=lustre-b1_8 clientbuildno=258 \  
serverjob=lustre-master serverbuildno=1270 \  
testgroup=full
```

# Autotest Test Parameters (contd.)

- failover testing with shared storage

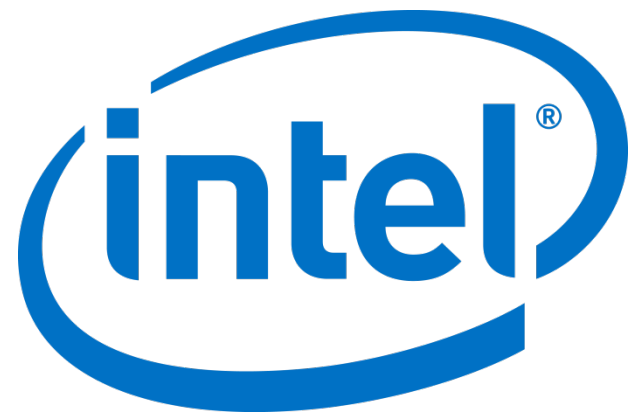
```
Test-Parameters: envdefinitions=SLOW=yes clientcount=4 \  
osscount=2 mdscount=2 austeroptions=-R failover=true \  
useiscsi=true testlist=recovery-mds-scale
```

- debugging a specific subtest

```
Test-Parameters: fortestonly \  
envdefinitions=ONLY=27,PTLDEBUG=-1,DEBUG_SIZE=100 \  
testlist=sanity
```

# References

- Test Framework Environment Variables
  - <http://wiki.whamcloud.com/display/PUB/Lustre+Test+Tools+Environment+Variables>
- Autotest Test Parameters
  - <http://wiki.whamcloud.com/display/PUB/Changing+Test+Parameters+with+Gerrit+Commit+Messages>



Thank You!