Setting up a Lustre Test Environment on a local system

Systems Description

- Three nodes in total
  - MGS: MRtest01
  - OSS: MRtest02
  - Client: MRtest03

Instructions

The goal of this instruction set is to describe how to setup tests to run on a local VM setup. The end target is to ease the debugging burden.

1. Install Lustre RPMs on all your test nodes
2. Install pdsh
   a. el6: https://build.hpdd.intel.com/job/toolkit/arch=x86_64,distro=el6/lastSuccessfulBuild/artifact/_topdir/RPMS/x86_64/
      i. pdsh-rcmd-ssh-2.29-1.wc1.x86_64.rpm
      ii. pdsh-2.29-1.wc1.x86_64.rpm
   b. el7: https://build.hpdd.intel.com/job/toolkit/arch=x86_64,distro=el7/lastSuccessfulBuild/artifact/_topdir/RPMS/x86_64/
      i. pdsh-rcmd-ssh-2.29-1.wc1.x86_64.rpm
      ii. pdsh-2.29-1.wc1.x86_64.rpm
3. Setup passwordless ssh login among all nodes:
   a. http://www.thegeekstuff.com/2008/11/3-steps-to-perform-ssh-login-without-password-using-ssh-keygen-ssh-copy-id/
   b. YOU HAVE TO make sure that the node which you run the tests on can ssh passwordlessly into itself
   c. Make sure to ssh into all the nodes to get rid of any prompts.
4. Insure that lustre-iokit* is installed on all the nodes
   a. yum install lustre-iokit-2.8.50-2.6.32.504.16.2.el6_lustre_g28cd67e.x86_64.rpm
5. Setup the /etc/hosts file to map hostname to ip address on all nodes.

   a.
```
127.0.0.1               localhost.localdomain localhost
192.168.122.199         MRtest01 MRtest01.localdomain
192.168.122.155         MRtest02 MRtest02.localdomain
192.168.122.218         MRtest03 MRtest03.localdomain
::1             localhost6.localdomain6 localhost6
```

6. Setup the /usr/lib64/lustre/tests/cfg/local.sh to reflect your setup on the client (where you'll be running the tests). Modify this file by adding the below at the top of the file. No need to change anything else in the file.

   a.
```
#file system name
FSNAME=lustrewt
# mount point on the client
MOUNT=/mnt/client
# hostname of the mds
mds_HOST=MRtest01
# hostname of the mds (repeated as above)
mds1_HOST=MRtest01
# mount point on the mds
mds1_MOUNT=/mnt/mdt
# number of MDSs in the test setup
MDSCOUNT=1
# physical device to formate using mkfs.lustre for the MDS
MDSDEV1=/dev/vdb
# number of OSTs in the system.
OSTCOUNT=1
# host name of the OST
ost_HOST=MRtest02
# host name of the OST (repeated as above)
ost1_HOST=MRtest02
# physical device to format using mkfs.lustre for the OST
OSTDEV1=/dev/vdb
# OST mount point
ost1_MOUNT=/mnt/ost
# PDSH command
PDSH="pdsh -S -Rssh -w"
```

7. Run llmount.sh
    a. go to /usr/lib64/lustre/tests/
    b. ./llmount.sh
    c. This will create and mount the file system. If this succeeds the filesystem should be in place.
8. Run a test (or subtest)
    a. cd /usr/lib64/lustre/tests/
    b. ./auster -v sanity.sh --only 0a
        i. You can add '-r' to auster to avoid calling llmount.sh
    c. you should see something like

```
[root@MRtest03 tests]# ./auster -v sanity.sh --only 0a
Started at Tue Apr  5 12:36:03 PDT 2016
MRtest03.localdomain: Checking config lustre mounted on /mnt/client
Checking servers environments
Checking clients MRtest03.localdomain environments
Logging to local directory: /tmp/test_logs/2016-04-05/123602
running: sanity.sh ONLY=0a
run_suite sanity /usr/lib64/lustre/tests/sanity.sh
-----============= acceptance-small: sanity ============----- Tue Apr  5 12:36:09 PDT 2016
Running: bash /usr/lib64/lustre/tests/sanity.sh
MRtest03.localdomain: Checking config lustre mounted on /mnt/client
Checking servers environments
Checking clients MRtest03.localdomain environments
Using TIMEOUT=20
disable quota as required
osd-ldiskfs.track_declares_assert=1
osd-ldiskfs.track_declares_assert=1
running as uid/gid/euid/egid 500/500/500/500, groups:
 [touch] [/mnt/client/d0_runas_test/f11819]
excepting tests: 76 42a 42b 42c 42d 45 51d 68b
skipping tests SLOW=no: 24o 24D 27m 64b 68 71 77f 78 115 124b 300o
preparing for tests involving mounts
mke2fs 1.42.13.wc4 (28-Nov-2015)
debug=-1
== sanity test 0a: touch; rm ======================== 12:36:18 (1459884978)
/mnt/client/f0a.sanity has type file OK
/mnt/client/f0a.sanity: absent OK
Resetting fail_loc on all nodes...done.
PASS 0a (2s)
resend_count is set to 4
resend_count is set to 4
resend_count is set to 4
resend_count is set to 4
resend_count is set to 4
== sanity test complete, duration 17 sec == 12:36:26 (1459884986)
debug=super ioctl neterror warning dlmtrace error emerg ha rpctrace vfstrace config console
lfsck
sanity.sh returned 0
Finished at Tue Apr  5 12:36:27 PDT 2016 in 25s
./auster: completed with rc 0
```

9. Auster usage help

a.
```
Usage auster [options]  suite [suite options] [suite [suite options]]
Run Lustre regression tests suites.
      -c CONFIG Test environment config file
      -d LOGDIR Top level directory for logs
      -D FULLLOGDIR Full directory for logs
      -f STR    Config name (cfg/<name>.sh)
      -g GROUP  Test group file (Overrides tests listed on command line)
      -S TESTSUITE First test suite to run allows for restarts
      -i N      Repeat tests N times (default 1). A new directory
                will be created under LOGDIR for each iteration.
      -k        Don't stop when subtests fail
      -R        Remount lustre between tests
      -r        Reformat (during initial configuration if needed)
      -s        SLOW=yes
      -v        Verbose mode
      -l        Send logs to the Maloo database after run
                  (can be done later by running maloo_upload.sh)
      -h        This help.
Suite options
These are suite specific options that can be specified after each suite on
the command line.
   suite-name  [options]
      --only LIST         Run only specific list of subtests
      --except LIST       Skip list of subtests
      --start-at SUBTEST  Start testing from subtest
      --stop-at SUBTEST   Stop testing at subtest
      --time-limit LIMIT  Don't allow this suite to run longer
                          than LIMT seconds. [UNIMPLEMENTED]
Example usage:
Run all of sanity and all of replay-single except for 70b with SLOW=y using
the default "local" configuration.
  auster -s sanity replay-single --except 70b
Run all tests in the regression group 5 times using large config.
  auster -f large -g test-groups/regression -i 5
```

Resources

Testing a Lustre filesystem

Lustre Test Tools Environment Variables

Test Variable Definitions

https://testing.hpdd.intel.com/test_logs/fd7e055a-f984-11e5-812a-5254006e85c2/show_text

Lustre Test Tools Environment Variables