Community testing of Lustre

Contributing to maloo testing, May 2022

James Simmons
Storage Systems Engineer
Oak Ridge National Laboratory

ORNL is managed by UT-Battelle LLC for the US Department of Energy
We need to decentralize testing

• Currently whamcloud does all reported testing
  – RHEL8 servers / clients
  – Sanity + sanity-Inet testing for Ubuntu / SUSE / ARM
  – So many versions tested !!!!

• Redhat + CentoOS crises

• ARM: Linaro nearly completed own Arm64 test cluster setup for full reporting

• Still a work in progress.
Scale of test system

- Rate of turn around for testing
  - 2.5 Hours for sanity.sh testing
  - 36 hours for full testing
- Recommend 5 server nodes + 3 clients for each testing session
  - Double for ZFS + ldiskfs testing
- Normally ~20 patches per day → 160 VM testing sessions
  - 256 GB allocated per VM instances
Jenkins Setup

• Download from [https://pkg.jenkins.io](https://pkg.jenkins.io) + Java JDK and install with rpm/ apt-get

• Create a jenkins user with home directory + password
  
  } Setup $HOME/.gitconfig with username@email

• Unlock master Jenkins node [http://localhost:8080](http://localhost:8080) with admin password

• Systemctl enable jenkins; systemctl start jenkins; systemctl status jenkins
  
  } Watch out for filewalls
Jenkins slave setup

- Two ways to manage slaves

  ```java
  java -jar slave.jar -jnIUrl http://yourbuildserver/computer/yourbuildnodename/slave-agent.jnlp
  ```

  - Don’t know how to use pipeline scripts with this approach

- Manage Jenkins → Manage Nodes → New Node

  - Fill out form

  - On slave select “Create Item”. Goto to Pipeline and add your pipeline script

    ```java
    node("Slavenode") {
      stage("build") {
        sh 'your script to build lustre'
      }
      stage("test") {
        sh 'your script to test lustre'
      }
    }
    ```
Basic Gerrit setup

• Based on top of git
  – Configure git for whamcloud.com
    ● https://wiki.lustre.org/Using_gerrit

• Install gerrit-review
  – apt-get install git-review / yum install git-review
  – easy_install pip; pip install git-review==1.21
  – git config -global gitreview.remote origin
  – git config -global gitreview.username “username”
  – git review -s in your tree, explore the options
Whamcloud gerrit setup

• Setup build environment for Lustre. Kernels and libraries.
  – Checkout gerrit lustre tree off of git.
• Install python and python-request
• Make sure whamcloud gerrit account is “HTTP password”
• Download from https://github.com/verygreen/lustretest/blob/master/*
  – gerrit_build and test_new.py, run_build.sh, GERRIT_AUTH.example
• Rename GERRIT_AUTH.example and update username and password
Gerrit script setup

- Edit `run_build.sh` to point to your kernel + lustre source location
- Edit `gerrit_buildpath.py`
  - `CHECKPATH_RUN` to point to `run_build.sh`
  - `BUILDER_TYPE` to label test type “Ubuntu 22.04”
  - Set `REVIEW_HISTORY` to current UNIX time
  - Change `self.post_enabled` to true if you want to post results
Arm64 External Builder and Testing Cluster - By Linaro

Test Infra Needed:
- Self setup Jenkins server and slave (Leverage Whamcloud Jenkins is also OK)
- Use Terraform for hardware management, call Linaro Arm64 OpenStack based cloud to provision the test vm clusters and setup the environment.
- Execute test with Auster locally, then upload the test consequence to Lustre Community Maloo DB
- Test Code Repo: https://github.com/Linaro/lustretest

Jenkins Jobs:
- Lustre Build Job (Arm64)
- Lustre Test Job (Arm64)

Test Suites coverage:
- LUSTRE_TEST_SUITE_1 = "sanity sanity-pfl"
- LUSTRE_TEST_SUITE_2 = "ost-pools replay-single conf-sanity"
- LUSTRE_TEST_SUITE_3 = "insanity mmp replay-ost-single sanity-dom sanity-flr sanity-hsm sanity-quota"
- LUSTRE_TEST_SUITE_4 = "lustre-rsync-test recovery-small sanity-scrub sanityyn"
- LUSTRE_TEST_SUITE_5 = "mds-survey replay-dual runtests sanity-lfsck sanity-sec"
- LUSTRE_TEST_SUITE_6 = "recovery-small sanity-sec sanity-selinux large-scale"
Lustre Build Job:

- Use `gerrit_buildpatch.py` to watch the change in upstream.
- Trigger the build process in Linaro Arm64 Jenkins Slave.
- Use `build-release.sh` which leverage Lustre “lbuild” script to execute the build job.
- Move the build RPMs to `https://uk.linaro.cloud/release` as the external Arm64 Lustre repo.
Lustre Test Job:

- **Provision**: Use Terraform -> OpenStack to provision test clusters, and use cloud-init to install Lustre RPM which is generated from Lustre Build Job, configure the test environment.
- **Node Init**: configure the ssh, hosts and generate Lustre test config.
- **Run Auster** and upload the data to Maloo DB.
- **All the script and Terraform templates has been published at [Linaro github](https://github.com/).**
Acknowledgements

This work was performed under the auspices of the U.S. DOE by Oak Ridge Leadership Computing Facility at ORNL under contract DE-AC05-00OR22725.