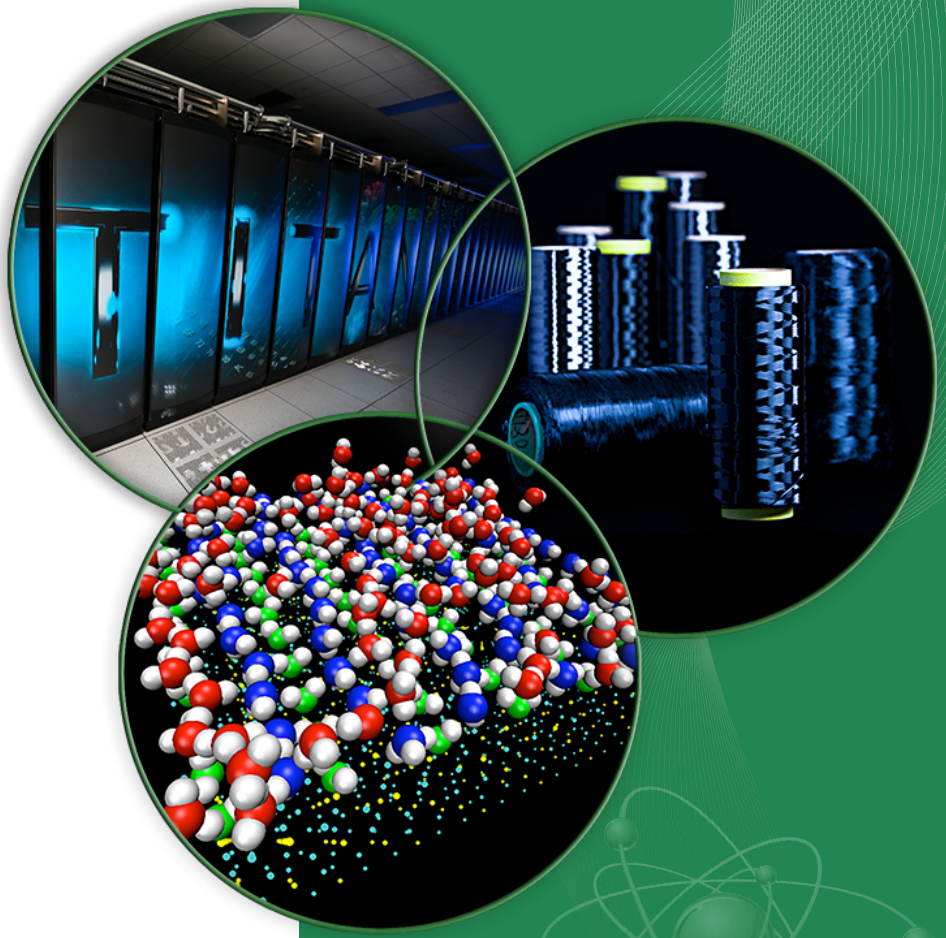
Oak Ridge National Laboratory
Computing and Computational Sciences Directorate

Evaluating the Functionality, Performance, and Reliability of Lustre



Jesse Hanley
Rick Mohr
Sarp Oral
Michael Brim
Nathan Grodowitz
Gregory Koenig
Jason Hill
Neena Imam

November 2015

ORNL is managed by UT-Battelle
for the US Department of Energy

OAK RIDGE National Laboratory

Overview

- Definition of terms
- Need and Reasoning
- Tools included in the Lustre Test Suite (LTS)
- OLCF Practices
- Benchmarking tools

Functionality

- Defined as “Testing based on an analysis of the specification of the functionality of a component or system”
- For example:
  - Setting a quota for a user or group
  - Testing to ensure the quota is enforced
- Does not include the code structure or implementation details

Performance

- Typically includes quantitative values such as how fast data can be written to or read from the file system
- Measurable metrics include:
  - Metadata operations such as file creates, stats, and deletions.
  - Small file IO performance
  - Streaming IO performance
  - Shared-file IO performance

Reliability

- Includes testing Lustre for failures and how those failures are handled
- These could include failures in the code, such as race conditions or edge cases
- Also includes failures in the underlying hardware and operating environment

The llmount.sh utility

- Note: llmount.sh and llmountcleanup.sh Bash scripts provided as part of the Lustre test suite
- Used to create or tear down a Lustre file system using loopback or block devices on a single node.
  - Settings are controlled with a configuration file and environment variables.
- Useful for testing functionality
- Use cases described on Intel's wiki
  - https://wiki.hpdd.intel.com/display/PUB/Testing+a+Lustre+filesystem

Auster

- Testing framework written in Bash provided by Lustre test suite
- New Lustre features and functionality should have a corresponding test in Auster
- Ability to run single and multi-node runs
- Patches submitted to Lustre must pass regression tests, which are run by Auster
- Austre logs and results can be automatically uploaded to Maloo, which is the database of test results

OLCF Initial Testing

- First, on development workstations:
  - The first tests include runs using the previously mentioned llmount.sh and Auster scripts
- Then, on technology integration testbed:
  - Tests begin to use real hardware – about 1/20th the scale of production
  - Not usually targeting performance
- Next, on production testbed:
  - Testbed is hardware that is similar to the production file system; same generation DDN SFA with the same pool configuration
  - This file system has the same upgrade history as the production file system
  - Same as the targeted environment will be, including software versions and tunings

OLCF – Production Testing (large-scale)

- Changes to Lustre software, configuration, or tuning are staged before outages
  - A new OS image is created for the nodes to boot into
  - Other changes are staged in configuration management
- Actual deployment of changes happens during dedicated testing periods
- Reliability testing includes forcing hardware and node failures
- Then, during this isolated period, run a set of benchmarks and system burn-in jobs
  - Results are compared to historical runs for regressions or potential improvements

Benchmarking Tools

File system related

- Included with Lustre:
  - LNet selftest
  - obdfilter-survey
  - sgpdd-survey
- Common software tools:
  - iozone
  - mdtest
  - IOR
  - simul

OLCF IO Harness

- Common user apps
- Includes ACME, HACC, NWCHEM, and XGC among others

Conclusion

- In order to test the impact of changes and upgrades, its important to be able to quantify those changes
- A well-defined test harness provides this functionality
- Lustre provides test suites
- Other various benchmarks are available

Resources

- http://wiki.lustre.org/Testing_HOWTO
- https://wiki.hpdd.intel.com/display/PUB/Testing+a+Lustre+filesystem
- http://wiki.lustre.org/Test_Descriptions
- http://www.tmmi.org/pdf/TMMi.Framework.pdf
- https://wiki.hpdd.intel.com/display/PUB/Using+Maloo
- http://www.eofs.eu/fileadmin/lad2014/slides/03_Shuichi_Ihara_Lustre_Metadata_LAD14.pdf
- http://www.eofs.eu/fileadmin/lad2014/slides/08_Roman_Grigoryev_Xperior__LAD14_Seagate.pdf
- http://www.eofs.eu/fileadmin/lad2015/slides/11_James_Numez_Lustre_Testing-_The_Basics.pdf

Acknowledgements

This work was supported by the United States Department of Defense (DoD) and used resources of the Computational Research and Development Programs at Oak Ridge National Laboratory.