Testing Lustre with Xperior

LAD, Sep 2014

Roman Grigoryev (roman.grigoryev@seagate.com)

Table of contents

Xperior: What is it?

- Xperior is a test execution framework
- The main test execution framework for Lustre testing in Seagate/Xyratex
- Currently supports execution of
  - Lustre tests (based on test-framework.sh)
  - LTP (linux test project)
  - LoadSim
  - draf support for IOR/MDTest
  - internal testing frameworks
- We use separate scripts for
  - Lustre installation
  - cluster restart logic
- No Xperior-native tests have been developed. But it is possible!

Xperior: What is it? Main features

1. Test management
   - exclude/include list from files and cli
   - possible options configuration per test
2. Execution control
   - executes on host which is not under test, increase predictability
   - conditional exit from test execution for restoration, e.g. for failed tests
   - random order for test execution
   - execute test many times
   - continue execution from last executed test
   - replay previously executed tests in original order
3. Test result reporting
   - own yaml-based format
   - tap report
   - html tap-based reporting + attachments
   - jenkins junit report + attachments
4. Extending behavior (see next slides)

Xperior: What is it? Lustre test support

Lustre tests - acceptance small/auster or test-framework.sh based tests

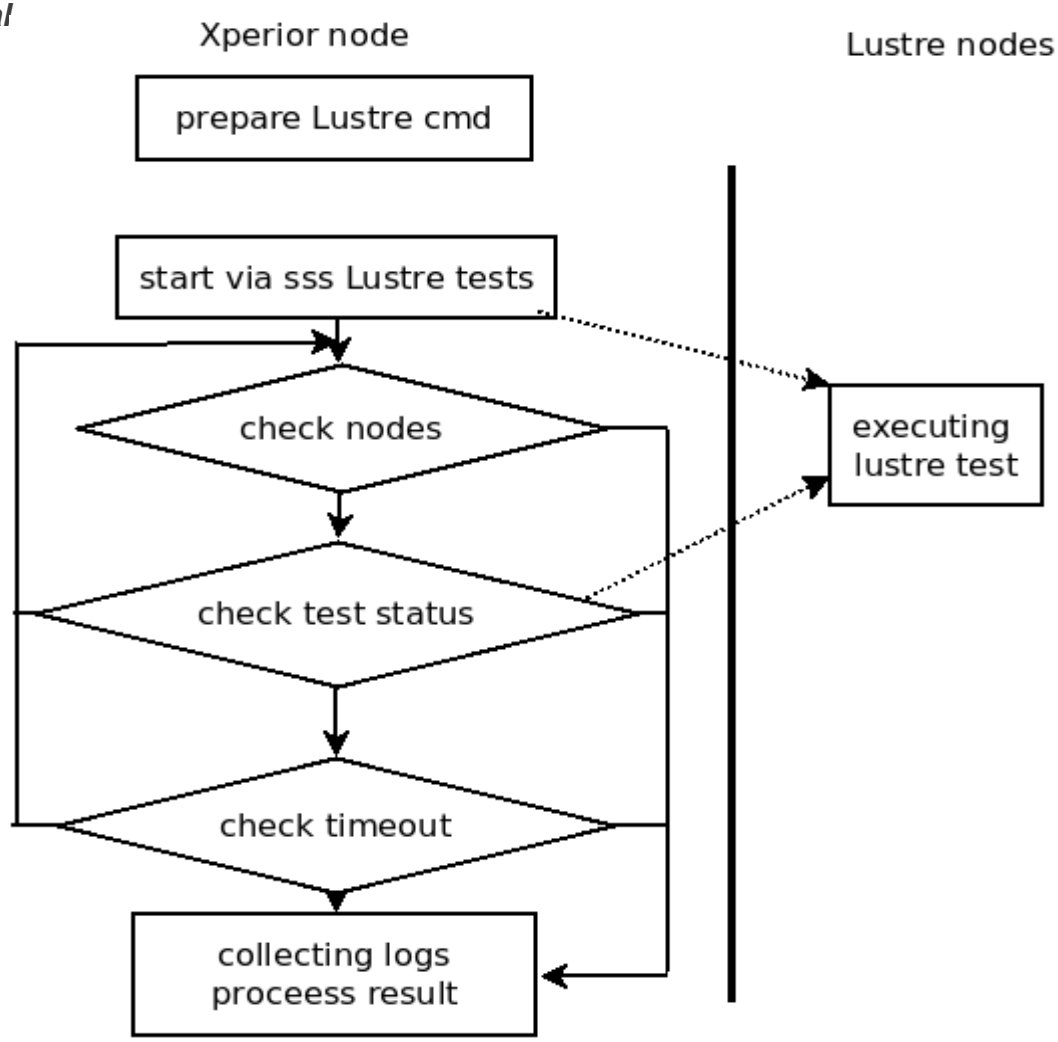
Approximate list of Lustre test suites executed by Xperior: ***conf-sanity, insanity, lustre-rsync-test, lustre-single, mmp, obdfilter-survey, ost-pools, racer, recovery-small, replay-dual replay-ost-single, replay-single, replay-vbr, sanity, sanity-benchmark, sanityn, large-scale, parallel-scale-nfsv3, parallel-scale, performance-sanity, flock-nfs, large-lun, sanity-hsm, parallel-scale-cifs, sanity-quota***

Supported cluster layouts

- 1 node
- 4 nodes (mds,oss, 2 clients)
- 1 mds (2 mdt), 1 oss (2 osts), 2 clients
- 8 nodes (2 mds: active-passive,2 oss: 6 osts active-active, 4 clients)
- Clusterstor CS9000

Supported special options

- client only mode
- ssh bridge between Xperior and Lustre nodes



Xperior: What is it? Sample Lustre tests configuration

Part of “testds/parallel-scale-openmpi-stress-hw_tests.yaml”

```
...
  - id: statahead
    timeout: 14400
dangerous: 'yes'
description: parallel scale Lustre tests
executor: Xperior::Executor::LustreTests
expected_time: 7200
groupname: parallel-scale-openmpi-stress-hw
script: parallel-scale.sh
reference: http://wiki.lustre.org/index.php/Testing_Lustre_Code
roles: StoreStat StoreSyslog StoreConsole GetDiagnostics StacktraceGenerator
schema: data/schemas/testds.yaml
tags: openmpi scale
timeout: 7200
env:
    cnt_DIR: /test-tools/cthon04
    cnt_NRUN: 20
    cbench_DIR: /test-tools/compilebench-0.6
    cbench_IDIRS: 10
    SIMUL: /test-tools/openmpi/simul-1.14/simul
    IOR: /test-tools/openmpi/ior-2.10.3/src/C/IOR
    ior_THREADS: 3
...
```

Xperior: What is it? extending test execution

Executors define how test will be executed.
Based on perl Moose extension
definition in test descriptor

```
executor: Xperior::Executor::LTPTests
```

List of public executors(`lib/Xperior/Executor/`):

Lustre.pm
- classic Lustre test-framework.sh-based tests executor

LustreSingleTests.pm
- specific Lustre test-framework.sh-based (e.g. lfsck) tests executor

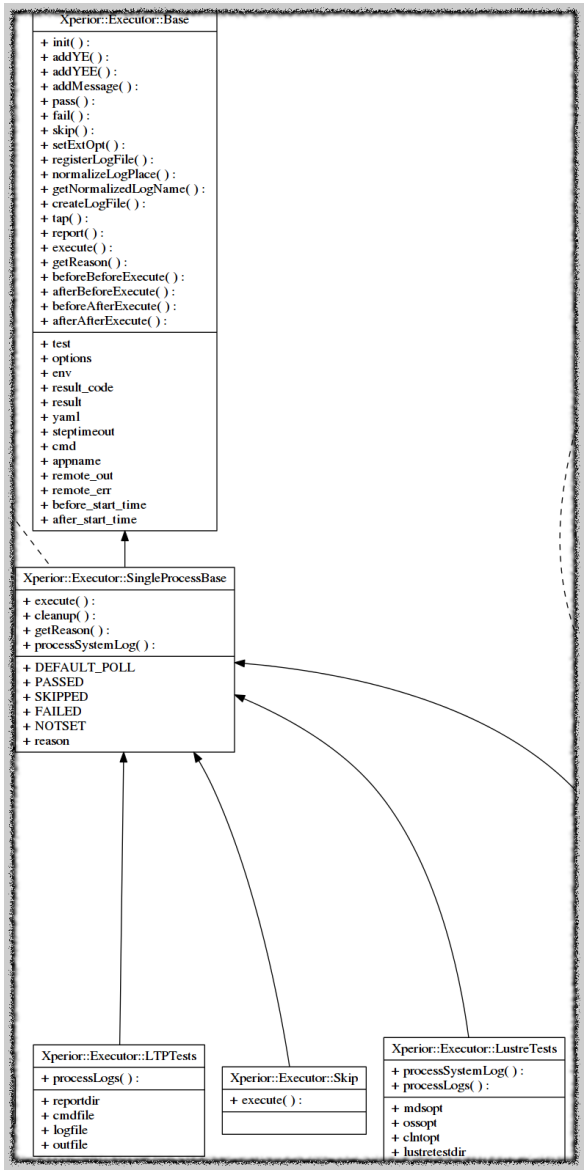
LTPTests.pm
- Linux Test Project, tests executor

LoadSimTests.pm
- LoadSim tests support, draft

IOR.pm/MDTest.pm
- IOR/MDTest tests support, draft

Noop.pm
- simple executor for testing needs

Skip.pm
- executor using dynamically mark as skipped, e.g. by exclude list



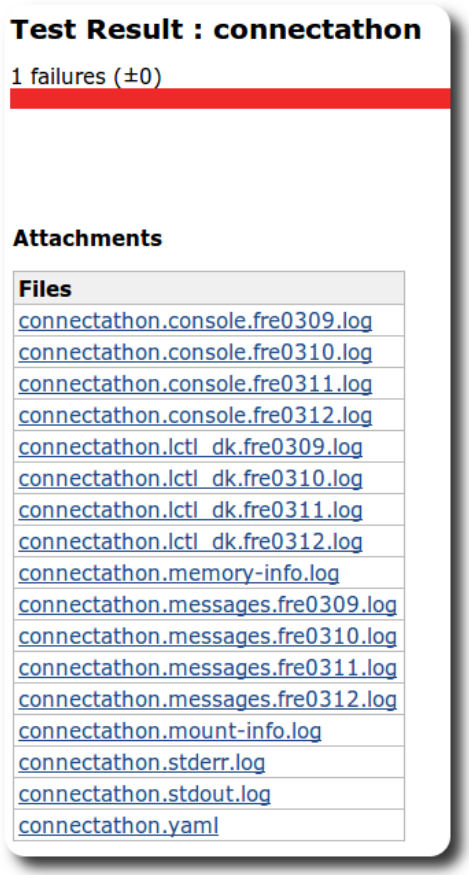
Xperior: What is it? extending behavior

Test execution behavior could be extended by writing Roles
Based on perl Moose::Role extension
definition test descriptor

```
roles: StoreStat StartMpdbootBefore StoreSyslog StoreConsole GetDiagnostics
  StacktraceGenerator
```

List of public roles(`lib/Xperior/Executor/Roles`):

StoreSyslog.pm - add /var/log/message to test result
GetDiagnostics.pm - executes Lustre script and to test result
NetconsoleCollector.pm - collect messages from upd receiver
ReformatBefore.pm - reformat before every test
StacktraceGenerator.pm - generate stracktrace when test fails
StartMpdbootBefore.pm - start mpd before executing test
StoreConsole.pm - add local file to test result, file descibes as console log
StoreStat.pm - collect some stat info before test execution
GetCoverage.pm - collect coverage infomation for every test, compatible only with 1-node setup, need special kernel



Xperior: How is it used?: Sample scenario

How we execute Lustre tests

- prepare test environment before testing - cobbler, kvm, puppet, private scripts
- Jenkins scripts for parallel execution by clusters:
  - configure nodes, install Lustre build for tests
  - prepare Xperior configuration
    - systemcfg.yaml
    - test descriptors
      - get lustre suite, e.g. sanity.sh from prepared system
      - generate new test descriptor based on template and suite file (bin/gentests.pl)
  - run xperior
  - if needed: catch failures, restart cluster, and restart xperior
- see results in jenkins

Xperior: Issues and limitations

- not well tested, we use only on limited setup number of setups
- no security
- low speed of test execution in comparing with acc-sm/auster
- 10-20 Lustre tests in master are not compatible with Xperior
- no Lustre/nodes maintenance (only nodes monitoring)
- perl :)

Xperior: Future plans

Implementation of Xperior native test platform

Possible basis:

- vfs (e.g. perl POSIX)
- LoadSim
- other ideas?

Required features:

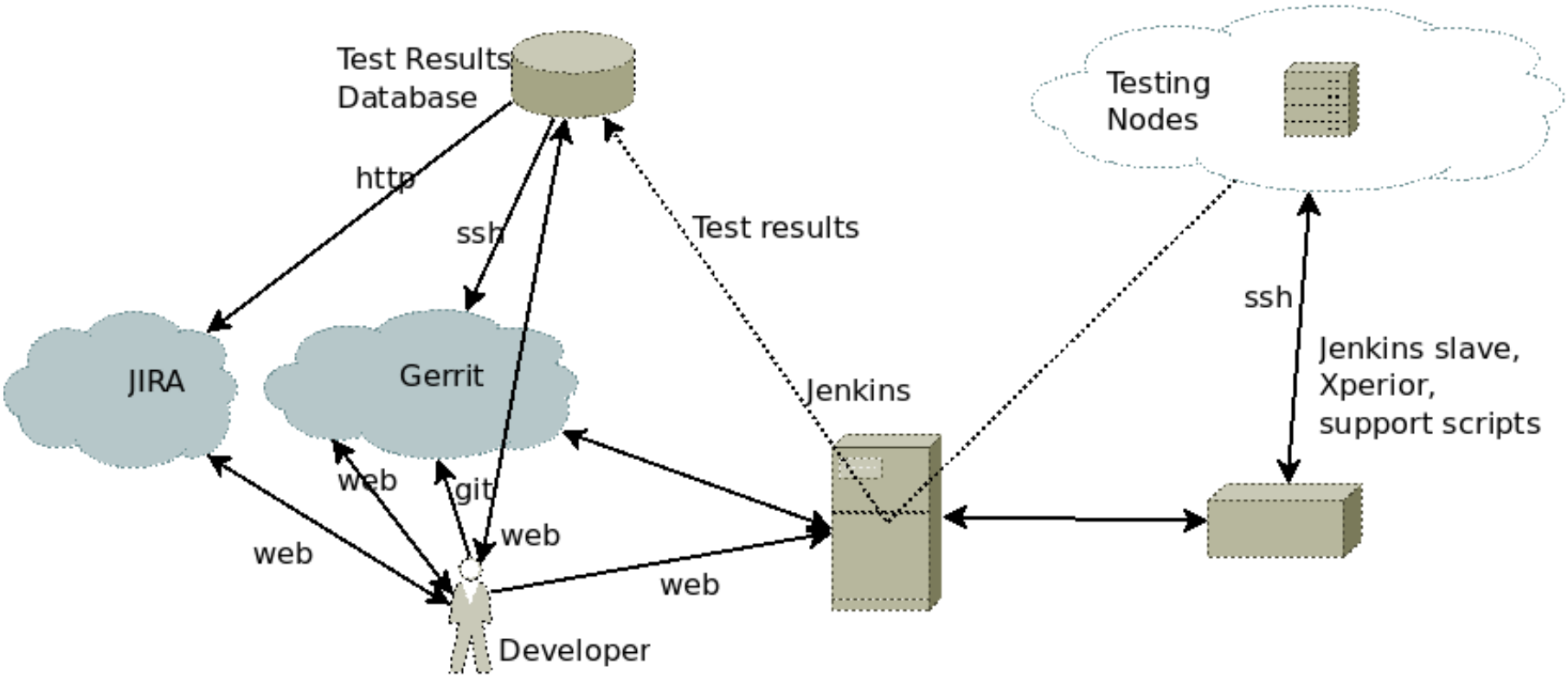
- distributed test support (MPI, Corba, events, ...)
- detection of node failure asap, correct test finish
- safe testing

Implementing simple agent (performance, control simplification)

Crash dump detection

Work issues from previous slide

CI: High level view



CI: High level view

1. integration testing
   - 4-node cluster (~2000 stored test session results), up to 10 testing session per day
   - kvm env, up to 24 clusters
   - expected test session execution is about 4h 15m
2. regular testing, include all supported clients
   - 1-nodes setup/4-nodes setup
   - exclude lists
3. special testing (include automated failure reproduction)
   - different clusters
4. testing on Clusterstor (CS9000, 1 storage)
   - up to 10 clients

CI: Jenkins and scripts

From Jenkins we use:

- parallelization based on slaves
- slave selection based on labels
- start options, execution console and junit report visualization
- integration with gerrit
- ordinary history tracking
- execution chain
- schedule re-execution
- cron-like scheduler

Scripts executed by Jenkins do

- kvm node stop/reset/start
- prepare Xperior configuration
- execute Xperior and observe Xperior’s status
- report to result database
- Lustre install/stop/start/mount

CI: Jenkins and scripts

quad02-quartet-2

Building integration-quartet » conf-sanity,quartet #679

quad02-quartet-3

Idle

quad03-quartet-1

Building integration-quartet » replay-single,quartet #680

quad03-quartet-2

Building integration-quartet » conf-sanity,quartet #680

quad03-quartet-3

Idle

quad12-quartet-1

Idle

quad12-quartet-2

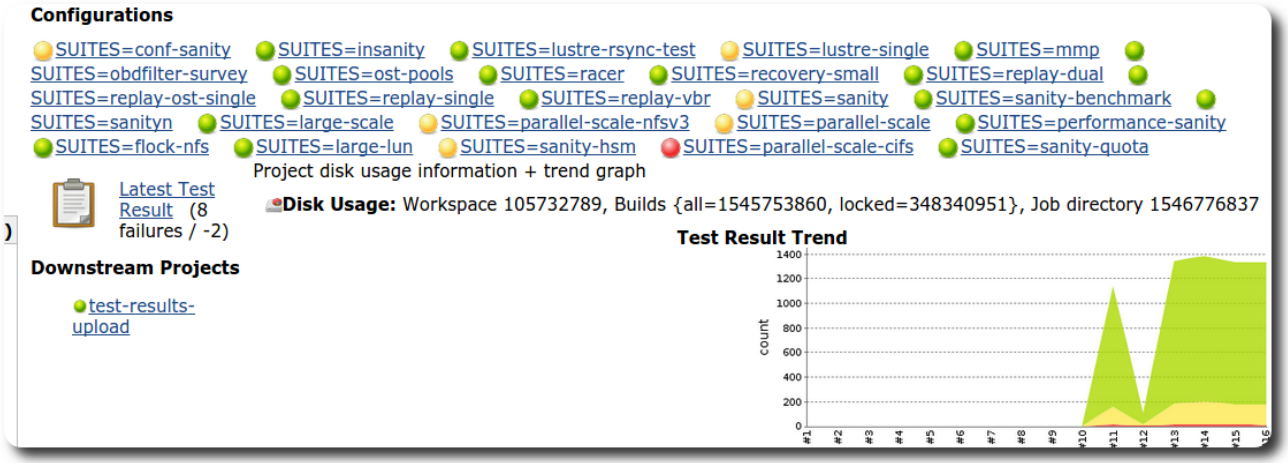
Idle

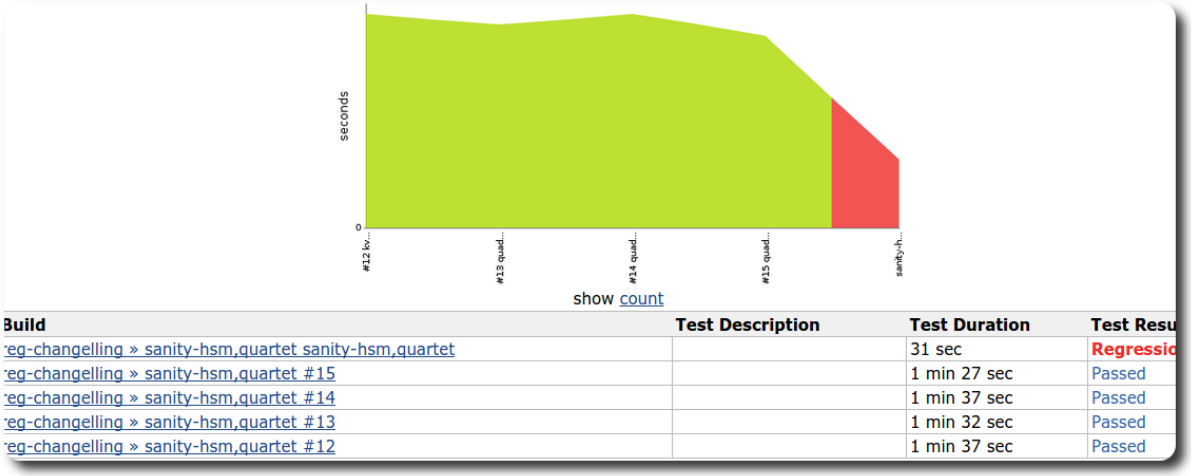
quad12-quartet-3

Idle

quad13-quartet-1

Building integration-quartet » sanity,quartet #680



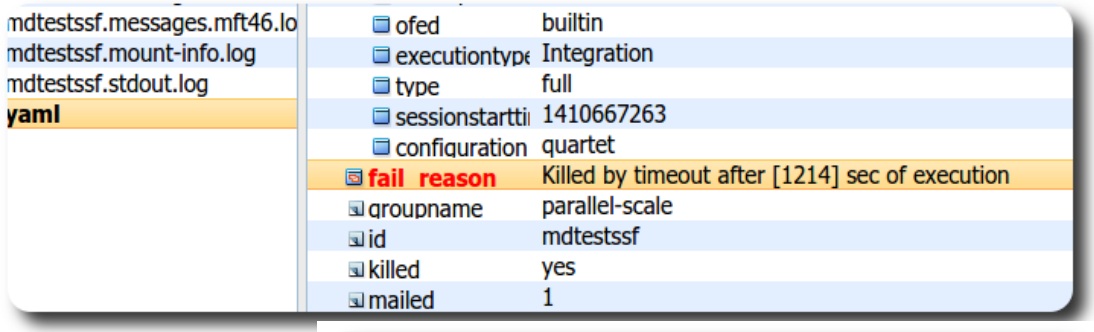


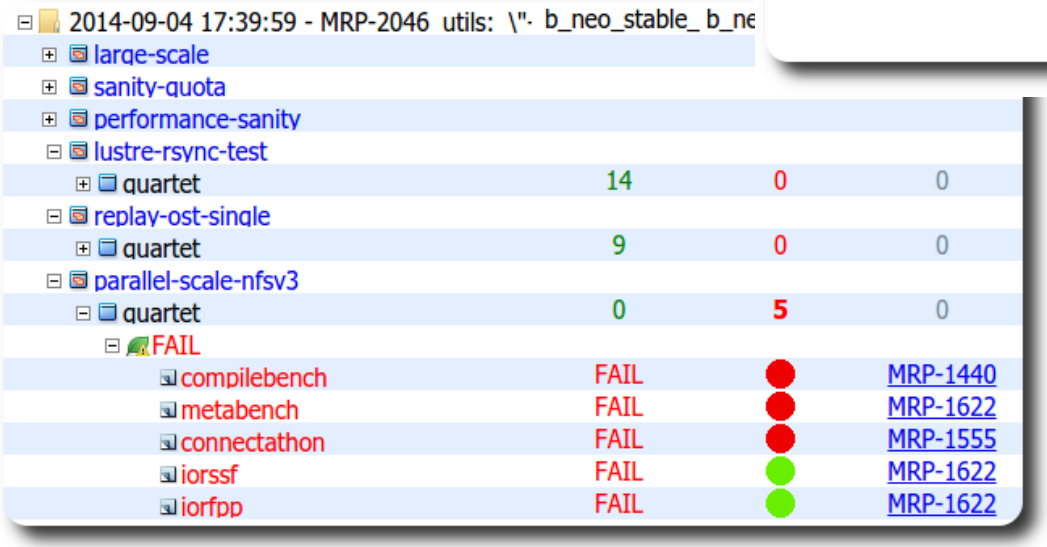
CI: Results database

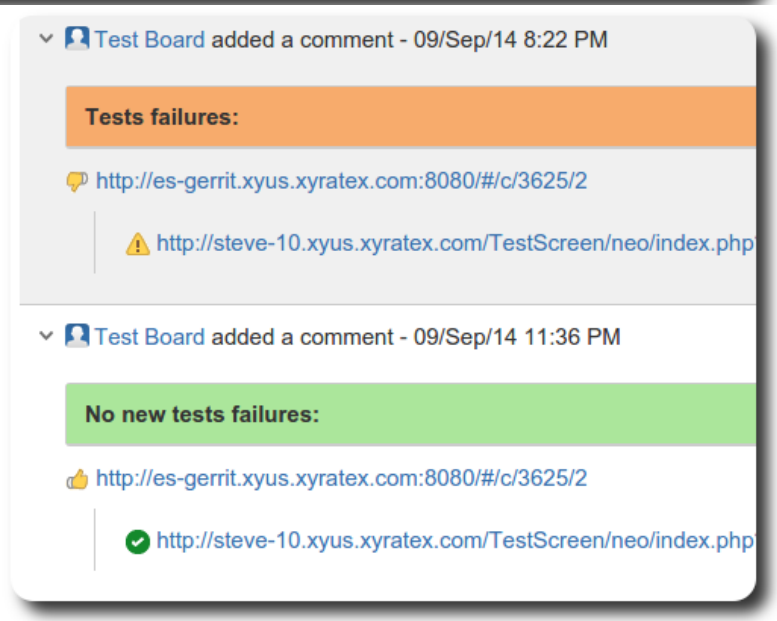
Results database - tool for digging into test results, simple vetting and linking with JIRA. Main features:

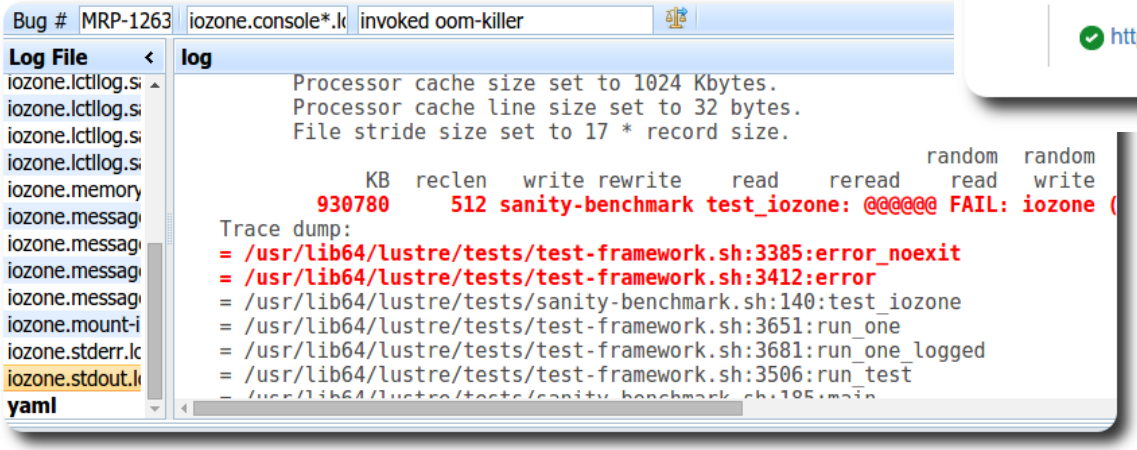
- filtering/searching by time, subject, branch
- access to all test logs
- associating log strings with failures
- associating failures with JIRA tickets
- compare test sessions
- schedule re-test
- works with xperior test result files with little modification
- written on mongo/php

CI: Results database









Testing and Amazon Web Services

We are looking AWS as platform for scale testing. We understand that it is not real hardware and not all issues will be found

Intel Cloud Edition for Lustre (Community Edition)
Some concerns (only for testing)

- not ready for testing (user, tools, so on)
- already has Lustre (for test we need to install our own)
- simple changes are difficult in cloudstack

Current setup

- 4-node classic setup (oss, mds, 2 clients, m3) + jenkins
- our testing images are SL6.5-based
- http://54.164.238.188/job/lustre_tests/

Price (approximately, based on current instances and workload)

- 4-x our cluster - ~$10 per day, ~$2.5 per day
- 8-x Community Edition cluster - ~$55 per day, ~$8 per day

Thank you all who participated in Xperior development and discussions!

Specially thanks to elena.gryaznova@seagate.com, alexander.lezhoev@seagate.com kyrylo.shatskyy@seagate.com

Xperior github

https://github.com/Xyratex/xperior

Xperior generated perldoc help

http://htmlpreview.github.io/?https://raw.githubusercontent.com/wiki/Xyratex/xperior/perldoc/index.html

Framework evaluation

http://wiki.opensfs.org/Automation_framework_evaluation

Framework requirements

http://wiki.opensfs.org/Test_framework_requirements

Questions?
Thank you!