

# Lustre User Group 2009

Quality Initiative  
**Robert Read**  
**Sun Microsystems**

# Overview

- Quality Initiative
- Current Situation
- Next Steps

# Quality Initiative

- How are we doing today?
- Where do we want to be in the future?
- Where do we start?

# QE Successes

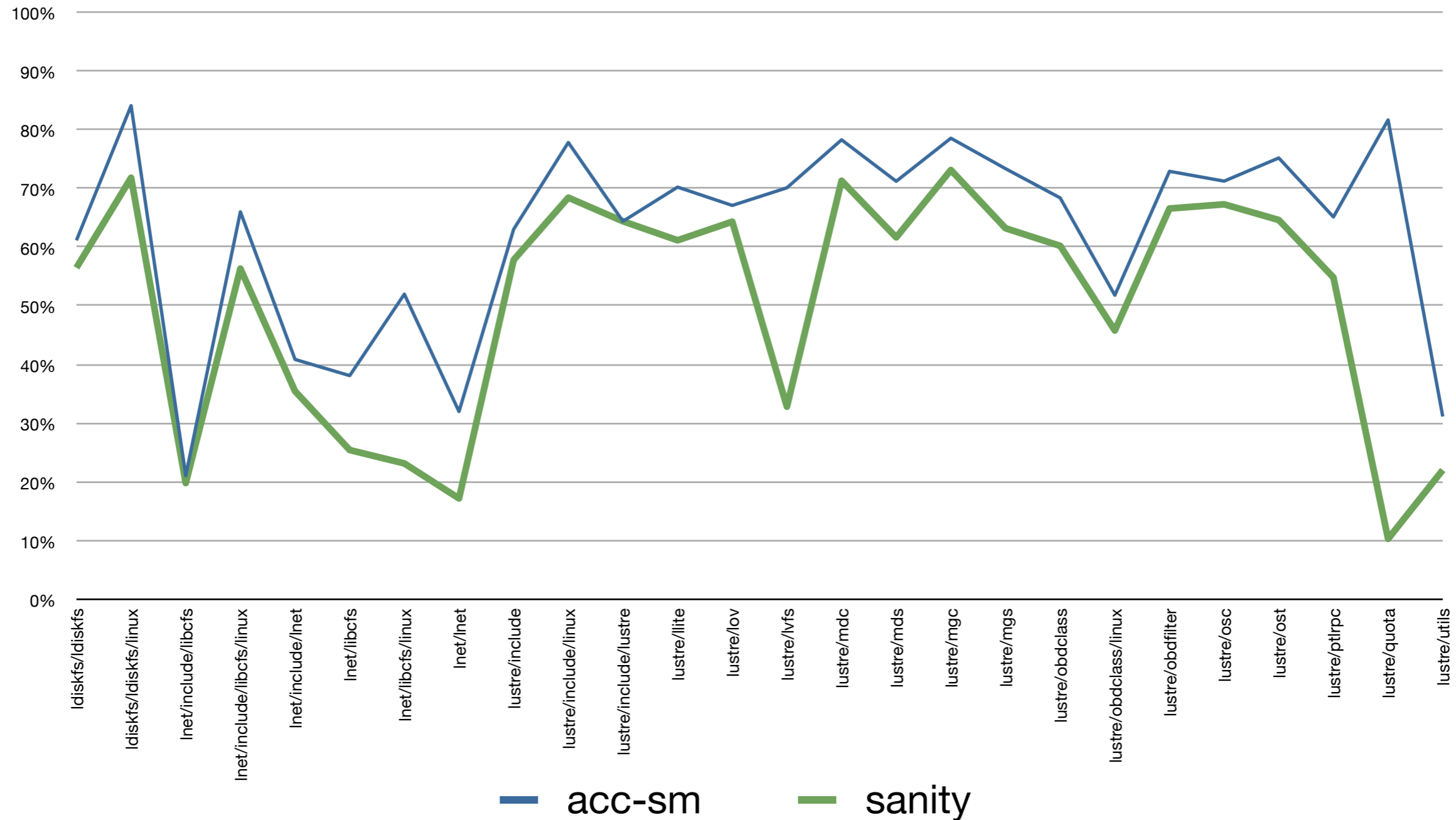
- LBATS - build automation on 4 architectures and OSs
- YALA - test automation
- Some Stage 2 testing automation
- Feature testing
- Found many bugs in our product

# Current Situation

# Existing coverage analysis

- Started basic coverage analysis
- `sanity.sh` on single node achieves 50% coverage overall
  - excluding `liblustre`, `libsysio`, `sockIn`, `Inet selftest`, etc
- 60-70% coverage of core Lustre modules

# sanity vs. acc-sm



# Customer reported issues

- As part of QI we have been talking to customers and partners
- Understand how they hit bugs that we missed
- Gathering feedback on test plans



# Top Customer Requests

- Failover/Recovery testing
- Realistic load testing (not just benchmarks)
- Large scale
- More feature testing
- Monitor performance regressions

# Internal Feedback

- Difficult to keep up with automated test results
- Many tests still run manually
  - Existing infrastructure not flexible enough
- Not enough resources to run all tests we want
  - Failover requires shared devices
- Need better tools to manage testing load

# Next Steps

# Short-term Goals

- Acquire more resources
- Additional feature testing
- Provide better tools for QE and developers
- Manage information
- Integrate testing with development process

# New Testing Resources

- Doubling size of automated test bed
- New cluster dedicated for HA testing
- Internal Sun resources
- Hyperion - LLNL's testbed

# More Feature Testing

- Continuing to add new tests added to acc-sm
  - specific feature tests (replay-vbr, sanity\_gss)
  - scale tests
- Detailed test plans being written

# Acceptance Small Todo List

- Integrate with standard config tools
- Capture detailed test results
- Collect more data (profiling, coverage, etc)
- Separate tests into Levels

# Post-run Analysis

- Save detailed test info searchable format (database)
- Compare test runs
  - find new failures
  - perf regressions
- Chop search to find regressions
- Update bugzilla from autovetted data



# Testing Automation

- Integrate manual release tests with acc-sm
- New tests added directly
- Allow all tests to be run externally

## 2.0 Development Process

- Short development cycles (~4 weeks)
- Focus on specific issues for each milestone
- Don't allow regressions (continuous testing)

# Summary

- Automation
  - Manage test result data
  - Easier to write and use
- Coverage
  - Understand our existing tests
  - Focus on real-world scenarios
- Improved Process



Thank You

Robert Read  
[rread@sun.com](mailto:rread@sun.com)