4.0 - Build and Test

Terrell Russell, Ph.D.
Renaissance Computing Institute (RENCI)
University of North Carolina at Chapel Hill

Motivation and Goals

Build and Test serves multiple purposes. The target audiences include core developers, external developers, users, users' managers, and grid administrators.

- Transparency (in both process and product)
- Use existing industry best practices
- Coverage -> Confidence in Refactoring
- Packaging -> Ease of installation and upgrade
- Test framework idempotency
- Test independence
- Topology awareness

- Automation, Automation, Automation

Background

- iRODS 3.x Perl-based test framework
  - Strengths
    - Provided use cases / example usage
    - Good coverage of the most used features
    - Helpful summary of results
    - Cross platform
    - Good for confirmation of success
  - Weaknesses
    - Not idempotent on test failure
    - Not independent
    - ~20% overall coverage
    - Hard to incorporate new tests (and use test driven development)

Today

- iRODS 4.0+ Python-based framework (uses unittest)
  - ~57% coverage

- Hudson – Continuous Integration
  - Polling based
  - Uses long-running VMs
  - Currently scheduling 187 independent jobs
  - Compiling with –Wall, -Werror, and –O3
  - Static analysis with cppcheck
  - Building packages
  - Testing packages

Today

- http://ci-dev.renci.org/hudson/view/iRODS
- Tests Currently in Continuous Integration
  - OS and Version
    - Ubuntu 10 and 12, CentOS 5 and 6, SuSE 11 and 12
  - Database
    - PostgreSQL, MySQL, Oracle
  - Microservice plugins
    - URL, MSO, workflows, etc.
  - Authentication plugins
    - Native, PAM, OSAuth
  - Network plugins
    - TCP, SSL
  - Resource plugins
    - Compound, Random, Replication, RoundRobin, Passthru, Deferred
    - Unixfilesystem, UnivMSS, S3, WOS, MockArchive, NonBlocking

Tomorrow

- Fully dynamic VM infrastructure
  - Build
  - Test
  - Deployed via JSON gridbundle descriptions of grid topologies, database backend, server locations, network speeds, policy/rules, and resource hierarchies.
  - Goal: Replicate any user-driven scenario or problem case
- Driven by post-commit hook
- Growing matrix of:
  - OS and Version
  - Database and Version
  - Resource Type
  - Feature Type
  - Topological Location

Motivation and Goals

Build and Test serves multiple purposes. The target audiences include core developers, external developers, users, users' managers, and grid administrators.

- Transparency (in both process and product)
- Use existing industry best practices
- Coverage -> Confidence in Tests and for Refactoring
- Packaging -> Ease of installation and upgrade
- Test framework idempotency
- Test independence
- Topology awareness

- Automation, Automation, Automation

Questions?

Terrell Russell, Ph.D.
Renaissance Computing Institute (RENCI)
University of North Carolina at Chapel Hill