Code Coverage and Continuous Integration

Presented at
Better Scientific Software tutorial

ECP 2nd Annual Meeting, Knoxville, Tennessee

Jared O'Neal
Argonne National Laboratory

NNSA National Nuclear Security Administration
U.S. DEPARTMENT OF ENERGY | Office of Science
IDEAS productivity

License, citation and acknowledgements

License and Citation

- This work is licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0).
- Requested citation: Jared O'Neal, Code Coverage and Continuous Integration, tutorial in Exascale Computing Project 2nd Annual Meeting, Knoxville, Tennessee, 2018. DOI: TBA.

Acknowledgements

- Alicia Klinvex
- This work was supported by the U.S. Department of Energy Office of Science, Office of Advanced Scientific Computing Research (ASCR), and by the Exascale Computing Project (17-SC-20-SC), a collaborative effort of the U.S. Department of Energy Office of Science and the National Nuclear Security Administration.
- This work was performed in part at the Argonne National Laboratory, which is managed by UChicago Argonne, LLC for the U.S. Department of Energy under Contract No. DE-AC02-06CH11357.

IDEAS productivity

ECP EXASCALE COMPUTING PROJECT

Code Coverage

How do we determine what other tests are needed?

Code coverage tools

- Expose parts of the code that aren't being tested
- gcov
  - standard utility with the GNU compiler collection suite
  - compile with --coverage
  - counts the number of times each statement is executed
- lcov
  - a graphical front-end for gcov
  - available at http://ltp.sourceforge.net/coverage/lcov.php
- Hosted servers (*e.g.* coveralls, codecov)
  - graphical visualization of results
  - push results to server through continuous integration server

Code coverage output

Overall Analysis

SOURCE FILES ON BUILD 45

LIST 2 CHANGED 0 SOURCE CHANGED 0 COVERAGE CHANGED 0

COVERAGE	Δ	FILE	LINES	RELEVANT	COVERED
74.39		src/functions/linear_fcn_class.f90	301	82	61
100.0		src/general/modulo_mod.f90	52	3	3

Detailed Analysis

```
265         ! Error distribution same for all x values
266         delta = S*Sxx - Sx*Sx
267         if (delta == 0.0_wp) then
268             ERRORMSG("Cannot do linear least-sqrs.  Divide by zero.")
269             stop
270         end if
271         delta_inv = 1.0_wp / delta
```

Online tutorial - https://github.com/amklinv/morpheus

Other example - https://github.com/jrdoneal/infrastructure

Code coverage is popular

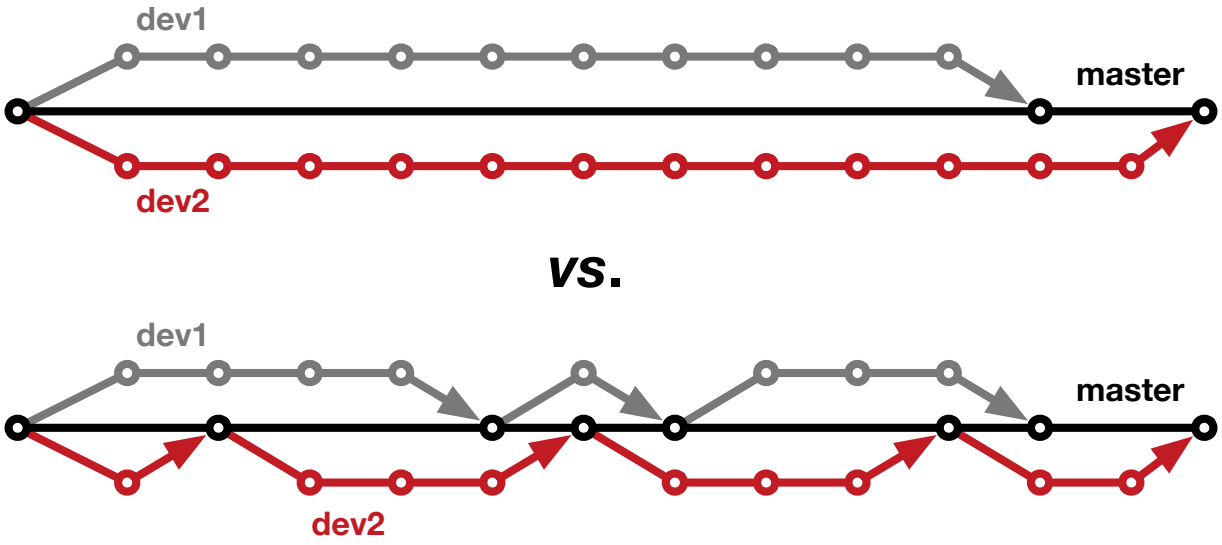
- gcov also works for C and Fortran
- Other tools exist for other languages
  - JCov for Java
  - Coverage.py for python
  - Devel::Cover for perl
  - profile for MATLAB
  - *etc*.

Continuous integration

The Short & Sweet of Continuous Integration

A master branch that always works

- Develop workflow policies
  - Commit and merge often
  - Test on push & pull requests?
- Code changes trigger automated builds/tests on target environments



IDEAS productivity

ECP EXASCALE COMPUTING PROJECT

Continuous integration (CI)

- Has existed for some time and interest is growing
  - ECP working with Travis CI to adapt for HPC machines
  - Dedicated Breakout Session

Thursday, February 8, 2018

10:30 AM-12:00 PM	**Breakout Sessions Track 1** **B-HI-9 | Ballroom B |** Continuous Integration Testing for ECP: An Essential Software Development Tool

- Setup, maintenance, and monitoring required
- Prerequisites
  - A reasonably automated build system
  - An automated test system with significant test coverage & useful feedback
  - Ability to bundle subset of tests
  - Builds/tests must finish in reasonable about of time

Cloud-based CI

- Linked to VCS hosts
  - GitHub & Travis CI
  - GitLab CI
  - BitBucket Pipelines
- Automated builds/tests triggered *via* pushes and pull requests
- Builds/tests can be run on cloud systems
- Test results are reported on the pull request page
- Can trigger code coverage analysis & documentation build
- Run tests on different environments

View of toy repository

https://github.com/jrdoneal/infrastructure

Repository Root

documentation	Included proper location of folders
src	Convert line_t into a class and impr
.gitignore	Travis CI should now run code cove
.travis.yml	Convert line_t into a class and impr
Makefile	Dont delete html or html/.git as this
README.md	Update README.md

README.md

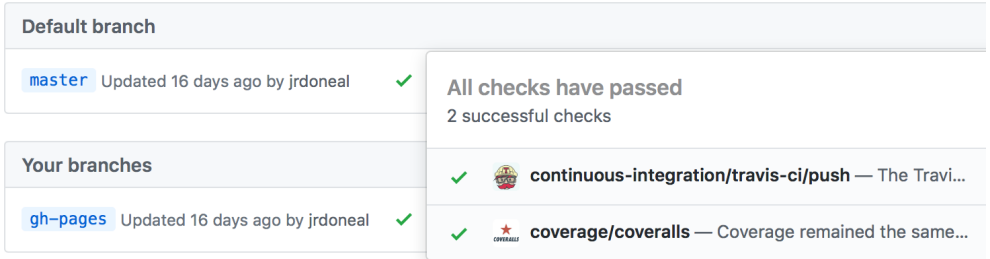
Sample .travis.yml

```
1  language: cpp
2
3  os:
4    - linux
5    - osx
6
7  compiler:
8    - g++
9    - clang
10
11 script: make && ./runtests.pl
```

Status of Personal codebase

Updated automatically

Results of CI Actions

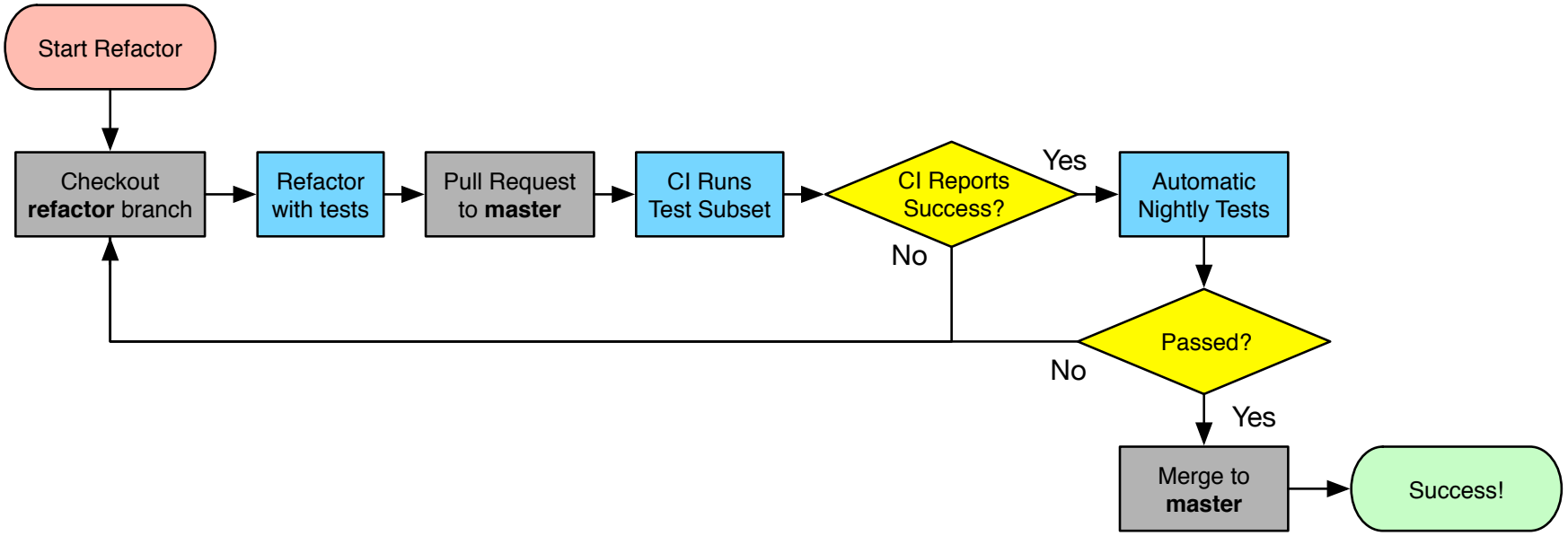


IDEAS productivity

ECP EXASCALE COMPUTING PROJECT

Putting it all together

Toy CI Workflow



IDEAS productivity

ECP EXASCALE COMPUTING PROJECT

Other resources

Software testing levels and definitions:
http://www.tutorialspoint.com/software_testing/software_testing_levels.htm

Working Effectively with Legacy Code, Michael Feathers. The legacy software change algorithm described in this book is very straight-forward and powerful for anyone working on a code that has insufficient testing.

Code Complete, Steve McConnell. Includes testing advice.

Organization dedicated to software testing: https://www.associationforsoftwaretesting.org/

Software Carpentry: http://katyhuff.github.io/python-testing/

Tutorial from Udacity: https://www.udacity.com/course/software-testing--cs258

Papers on testing:
http://www.sciencedirect.com/science/article/pii/S0950584914001232
https://www.researchgate.net/publication/264697060_Ongoing_verification_of_a_multiphysics_community_code_FLASH

Resources for Trilinos testing:
Trilinos testing policy: https://github.com/trilinos/Trilinos/wiki/Trilinos-Testing-Policy
Trilinos test harness: https://github.com/trilinos/Trilinos/wiki/Policies--%7C-Testing

Agenda

Time	Topic	Speaker
1:30pm-2:15pm	Why effective software practices are essential for CSE projects	Anshu Dubey, ANL
2:15pm-2:45pm	Better (small) scientific software teams	Michael A. Heroux, SNL
2:45pm-3:00pm	Improving Reproducibility Through Better Software Practices	Michael A. Heroux, SNL
3:00pm-3:30pm	Break	
3:30pm-4:15pm	Testing HPC Scientific Software: Introduction	Jared O'Neal, ANL
4:15pm-4:45pm	Verification, and Evaluating Project Testing Needs	Anshu Dubey, ANL
4:45am-5:00pm	Code Coverage and CI	Jared O'Neal, ANL