FOSDEM'20

Buildtest: HPC Software Stack Testing Framework

Shahzeb Siddiqui (Shahzeb.Siddiqui@3ds.com)

Dassault Systemes

FOSDEM'20

02/02/2020

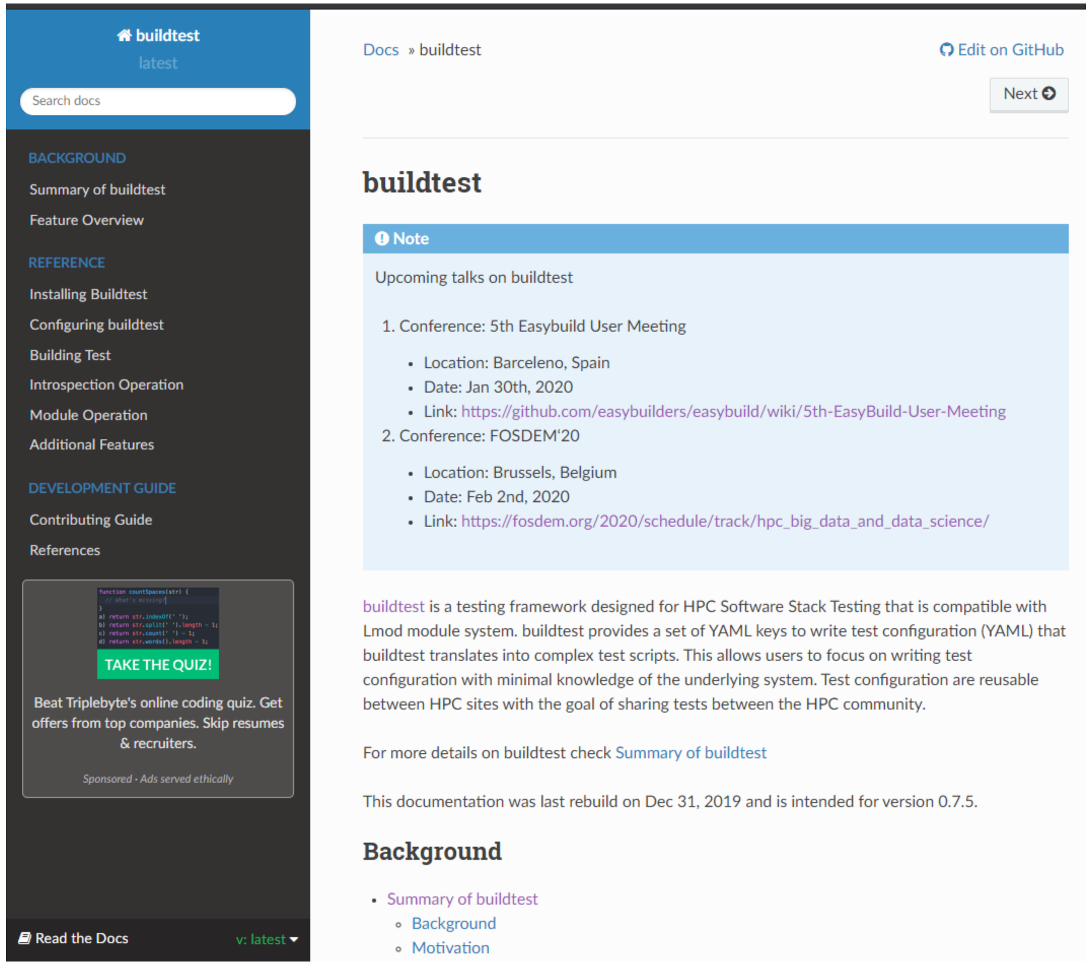
GitHub: https://github.com/HPC-buildtest/buildtest-framework
Documentation: http://buildtest.rtfd.io

Motivation

- Framework Requirements:
  - The framework is capable of testing of installed software in HPC Software Stack
  - The framework is able to integrate with module system
  - The framework provides users with a markup language for writing tests
  - The framework is able to automate test creation and execution
  - The framework provides a test repository that is community driven
- Buildtest is not meant to replace tools like make, cmake, or autoconf

What is buildtest

- Buildtest is a framework that:
  - Automates test script creation
  - Abstracts test complexity by using test configuration written in YAML
  - Allows Portable test configurations
  - Provides many module operations
- Buildtest comes with a repository of test configuration and source files



buildtest

Note

Upcoming talks on buildtest

1. Conference: 5th Easybuild User Meeting
   - Location: Barceleno, Spain
   - Date: Jan 30th, 2020
   - Link: https://github.com/easybuilders/easybuild/wiki/5th-EasyBuild-User-Meeting
2. Conference: FOSDEM'20
   - Location: Brussels, Belgium
   - Date: Feb 2nd, 2020
   - Link: https://fosdem.org/2020/schedule/track/hpc_big_data_and_data_science/

buildtest is a testing framework designed for HPC Software Stack Testing that is compatible with Lmod module system. buildtest provides a set of YAML keys to write test configuration (YAML) that buildtest translates into complex test scripts. This allows users to focus on writing test configuration with minimal knowledge of the underlying system. Test configuration are reusable between HPC sites with the goal of sharing tests between the HPC community.

For more details on buildtest check Summary of buildtest

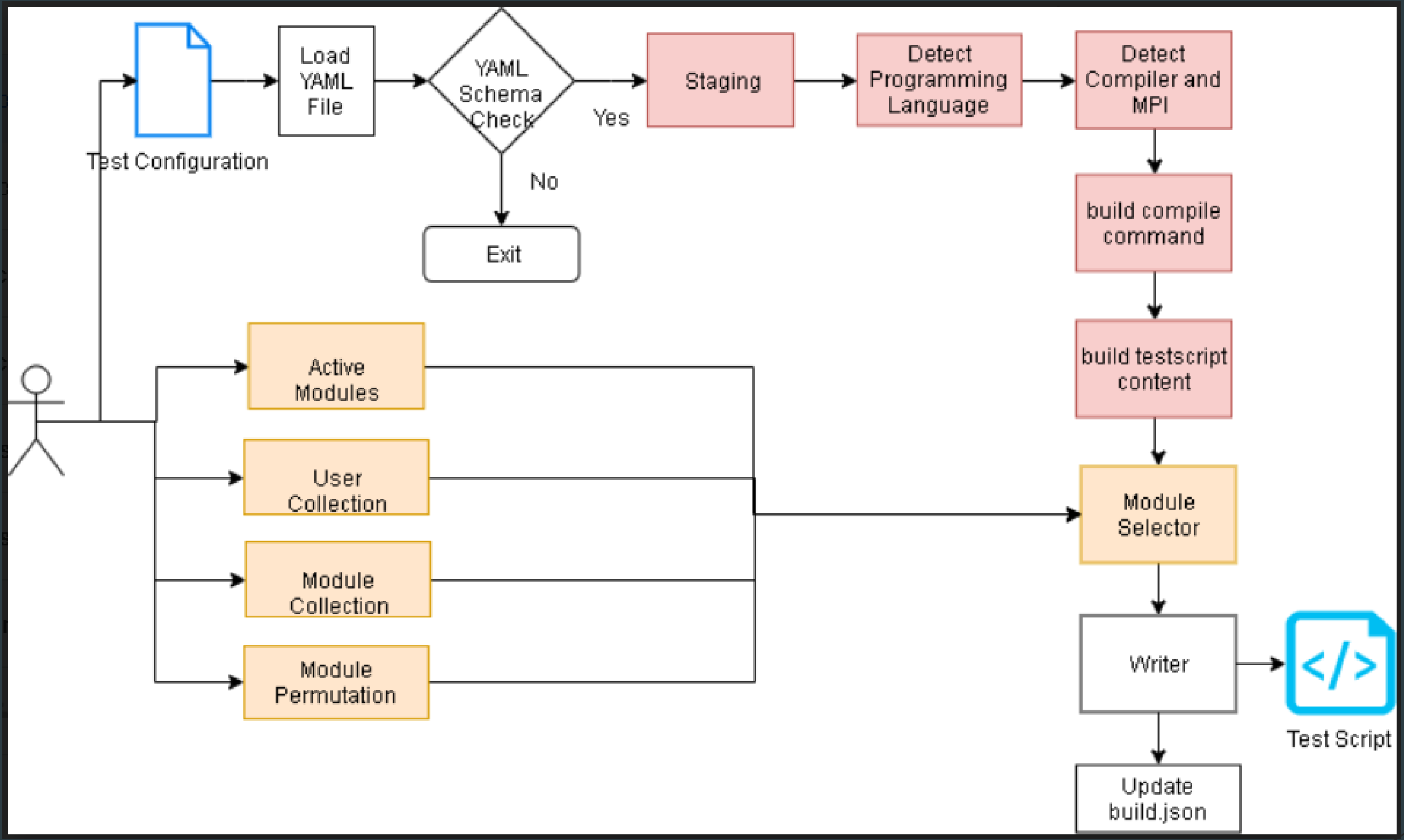
This documentation was last rebuild on Dec 31, 2019 and is intended for version 0.7.5.

Background

- Summary of buildtest
  - Background
  - Motivation

GitHub: https://github.com/HPC-buildtest/buildtest-framework
Documentation: http://buildtest.rtfd.io

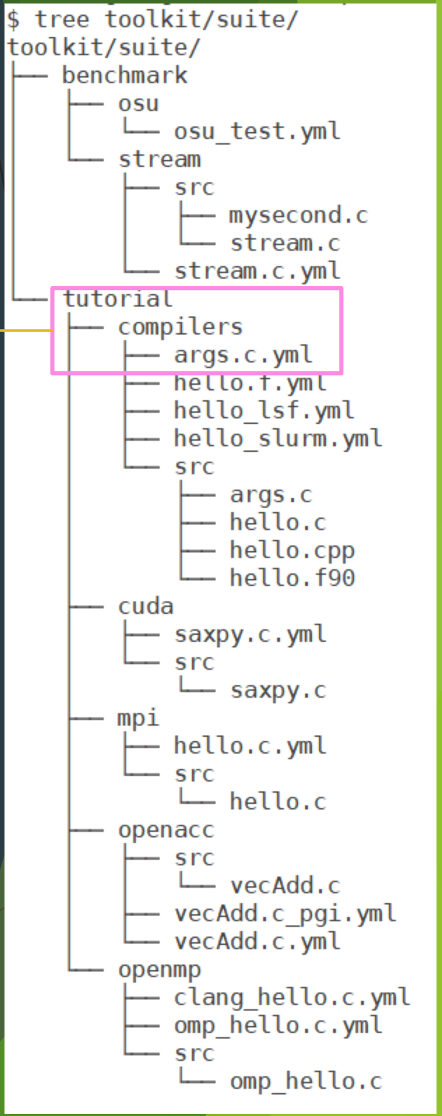
Build Pipeline



GitHub: https://github.com/HPC-buildtest/buildtest-framework
Documentation: http://buildtest.rtfd.io

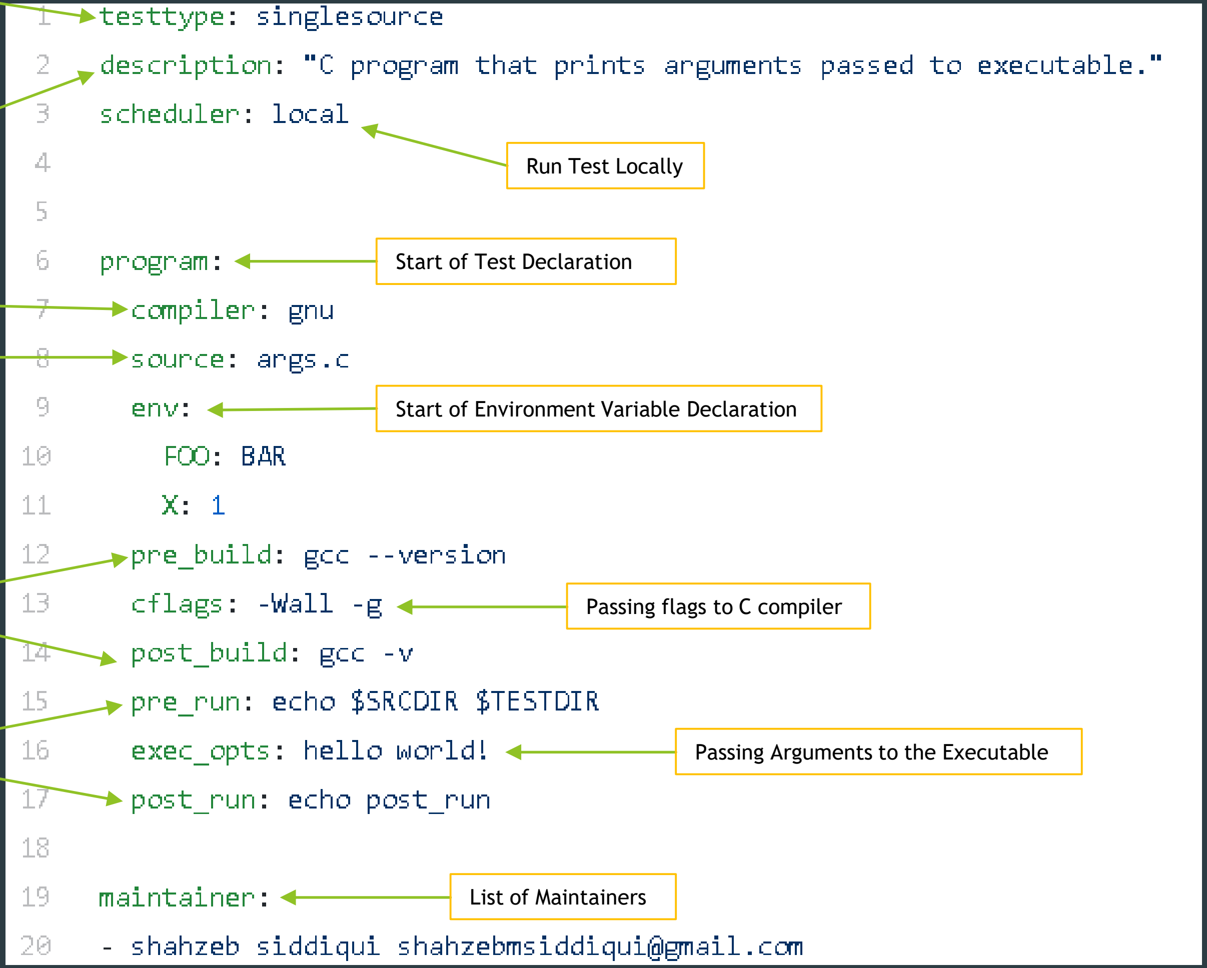
Building a Test

- To build a test script just specify a test configuration to buildtest as follows: buildtest build -c <test-configuration>
- The test configuration can be found under $BUILDTEST_ROOT/toolkit/suite
- Name of test configuration is formulated by replacing file separator (/) by a dot (.) so tutorial/compilers/args.c.yml → tutorial.compilers.args.c.yml
- Source code must be under src directory and test configuration must be named with extension .yml



```
$ tree toolkit/suite/
toolkit/suite/
├── benchmark
│   ├── osu
│   │   └── osu_test.yml
│   └── stream
│       ├── src
│       │   ├── mysecond.c
│       │   └── stream.c
│       └── stream.c.yml
└── tutorial
    ├── compilers
    │   ├── args.c.yml
    │   ├── hello.f.yml
    │   ├── hello_lsf.yml
    │   ├── hello_slurm.yml
    │   └── src
    │       ├── args.c
    │       ├── hello.c
    │       ├── hello.cpp
    │       └── hello.f90
    ├── cuda
    │   ├── saxpy.c.yml
    │   └── src
    │       └── saxpy.c
    ├── mpi
    │   ├── hello.c.yml
    │   └── src
    │       └── hello.c
    ├── openacc
    │   ├── src
    │   │   └── vecAdd.c
    │   ├── vecAdd.c_pgi.yml
    │   └── vecAdd.c.yml
    └── openmp
        ├── clang_hello.c.yml
        ├── omp_hello.c.yml
        └── src
            └── omp_hello.c
```

Test Configuration



```
testtype: singlesource
description: "C program that prints arguments passed to executable."
scheduler: local


program:
  compiler: gnu
  source: args.c
  env:
    FOO: BAR
    X: 1
  pre_build: gcc --version
  cflags: -Wall -g
  post_build: gcc -v
  pre_run: echo $SRCDIR $TESTDIR
  exec_opts: hello world!
  post_run: echo post_run

maintainer:
- shahzeb siddiqui shahzebmsiddiqui@gmail.com
```

Informs buildtest this is a Single Source Compilation. Implemented as a Python Class

Description of text. Limited to 80 chars

Run Test Locally

Start of Test Declaration

Specify Compiler Name

Source File to be compiled

Start of Environment Variable Declaration

Commands to run before and after compilation.

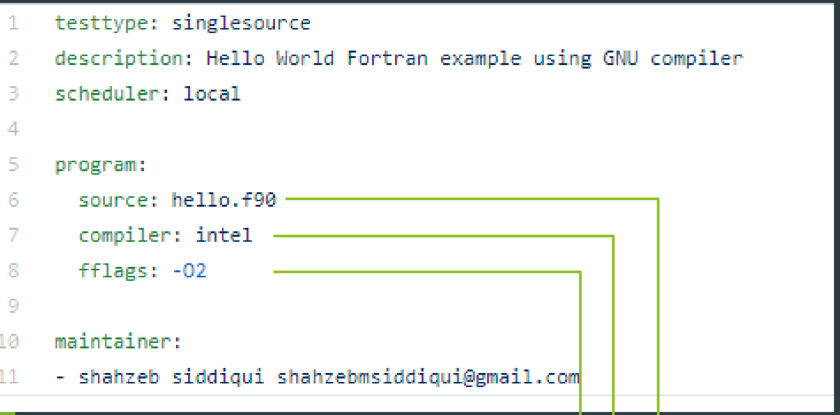
Passing flags to C compiler

Commands to run before and after execution.

Passing Arguments to the Executable

List of Maintainers

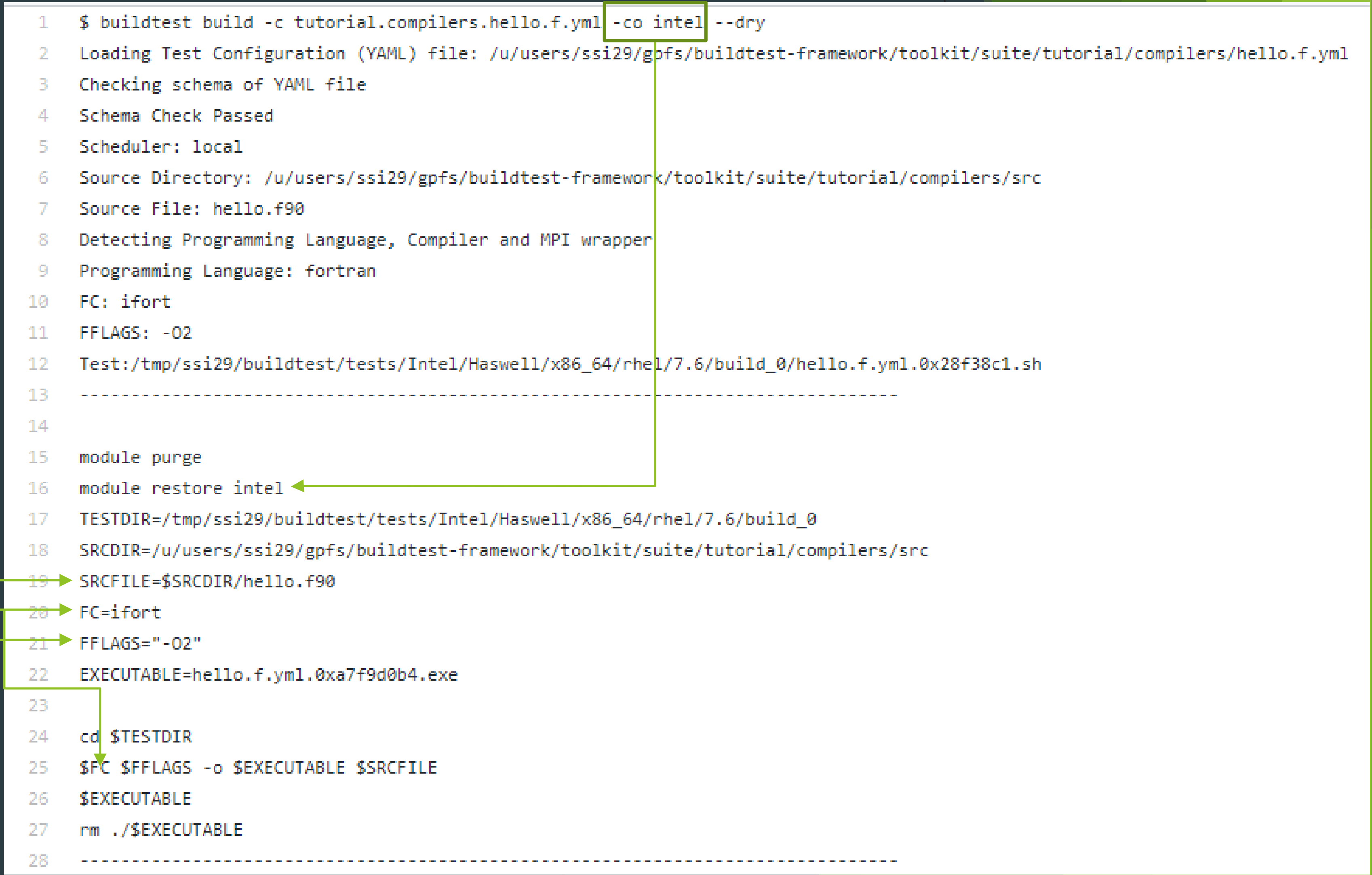
Intel Example



```yaml
testtype: singlesource
description: Hello World Fortran example using GNU compiler
scheduler: local

program:
  source: hello.f90
  compiler: intel
  fflags: -O2

maintainer:
- shahzeb siddiqui shahzebmsiddiqui@gmail.com
```

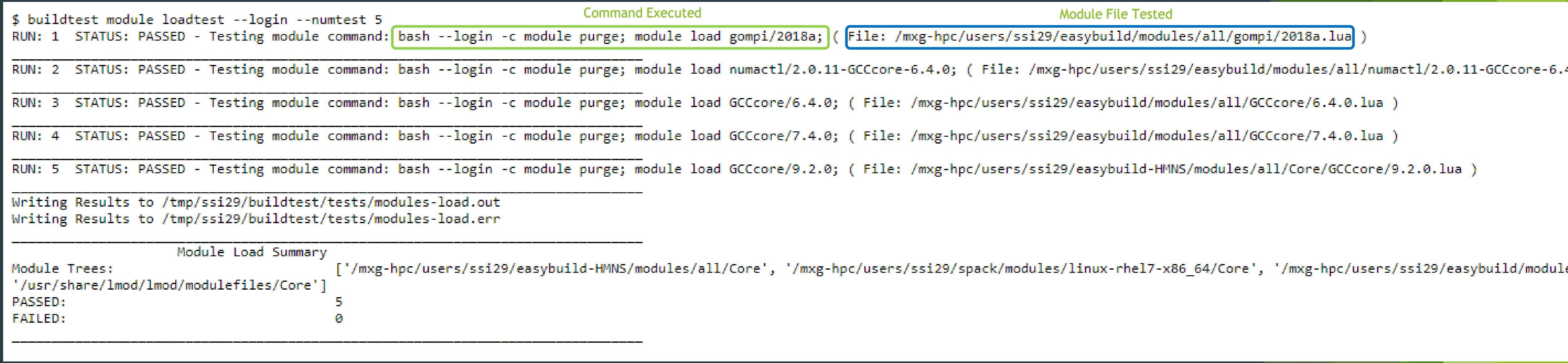


```
$ buildtest build -c tutorial.compilers.hello.f.yml -co intel --dry
Loading Test Configuration (YAML) file: /u/users/ssi29/gpfs/buildtest-framework/toolkit/suite/tutorial/compilers/hello.f.yml
Checking schema of YAML file
Schema Check Passed
Scheduler: local
Source Directory: /u/users/ssi29/gpfs/buildtest-framework/toolkit/suite/tutorial/compilers/src
Source File: hello.f90
Detecting Programming Language, Compiler and MPI wrapper
Programming Language: fortran
FC: ifort
FFLAGS: -O2
Test:/tmp/ssi29/buildtest/tests/Intel/Haswell/x86_64/rhel/7.6/build_0/hello.f.yml.0x28f38c1.sh
--------------------------------------------------------------------------------


module purge
module restore intel
TESTDIR=/tmp/ssi29/buildtest/tests/Intel/Haswell/x86_64/rhel/7.6/build_0
SRCDIR=/u/users/ssi29/gpfs/buildtest-framework/toolkit/suite/tutorial/compilers/src
SRCFILE=$SRCDIR/hello.f90
FC=ifort
FFLAGS="-O2"
EXECUTABLE=hello.f.yml.0xa7f9d0b4.exe

cd $TESTDIR
$FC $FFLAGS -o $EXECUTABLE $SRCFILE
$EXECUTABLE
rm ./$EXECUTABLE
--------------------------------------------------------------------------------
```

Module Load Testing

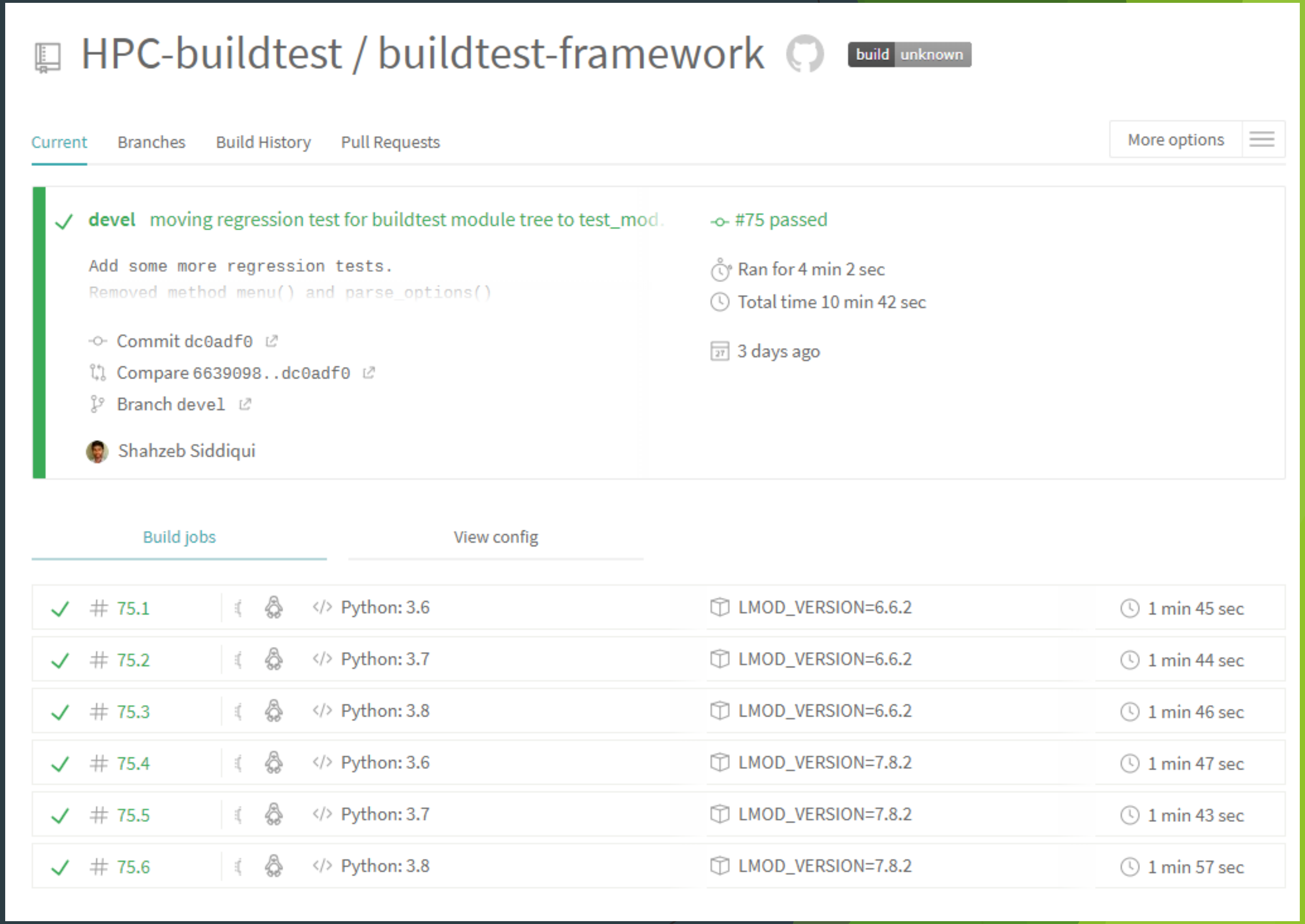


```
$ buildtest module loadtest --login --numtest 5
RUN: 1  STATUS: PASSED - Testing module command: bash --login -c module purge; module load gompi/2018a; ( File: /mxg-hpc/users/ssi29/easybuild/modules/all/gompi/2018a.lua )
________________________________________________________________________________
RUN: 2  STATUS: PASSED - Testing module command: bash --login -c module purge; module load numactl/2.0.11-GCCcore-6.4.0; ( File: /mxg-hpc/users/ssi29/easybuild/modules/all/numactl/2.0.11-GCCcore-6.4
________________________________________________________________________________
RUN: 3  STATUS: PASSED - Testing module command: bash --login -c module purge; module load GCCcore/6.4.0; ( File: /mxg-hpc/users/ssi29/easybuild/modules/all/GCCcore/6.4.0.lua )
________________________________________________________________________________
RUN: 4  STATUS: PASSED - Testing module command: bash --login -c module purge; module load GCCcore/7.4.0; ( File: /mxg-hpc/users/ssi29/easybuild/modules/all/GCCcore/7.4.0.lua )
________________________________________________________________________________
RUN: 5  STATUS: PASSED - Testing module command: bash --login -c module purge; module load GCCcore/9.2.0; ( File: /mxg-hpc/users/ssi29/easybuild-HMNS/modules/all/Core/GCCcore/9.2.0.lua )
________________________________________________________________________________
Writing Results to /tmp/ssi29/buildtest/tests/modules-load.out
Writing Results to /tmp/ssi29/buildtest/tests/modules-load.err
________________________________________________________________________________
                     Module Load Summary
Module Trees:                            ['/mxg-hpc/users/ssi29/easybuild-HMNS/modules/all/Core', '/mxg-hpc/users/ssi29/spack/modules/linux-rhel7-x86_64/Core', '/mxg-hpc/users/ssi29/easybuild/module
'/usr/share/lmod/lmod/modulefiles/Core']
PASSED:                                  5
FAILED:                                  0
________________________________________________________________________________
```

GitHub: https://github.com/HPC-buildtest/buildtest-framework
Documentation: http://buildtest.rtfd.io

Travis

- Since v0.7.4, buildtest can run its regression test in Travis. Several improvement to Travis configuration in v0.7.5
- Currently, buildtest contains approximately 30+ regression tests
- Some regression tests rely on having a software stack, so buildtest builds a mini stack using easybuild.
- Buildtest is tested for Python 3.6, 3.7, 3.8 and Lmod version 6.6.2 and 7.8.2


HPC-buildtest / buildtest-framework

✓ devel moving regression test for buildtest module tree to test_mod

```
Add some more regression tests.
Removed method menu() and parse_options()
```

Commit dc0adf0
Compare 6639098..dc0adf0
Branch devel
Shahzeb Siddiqui

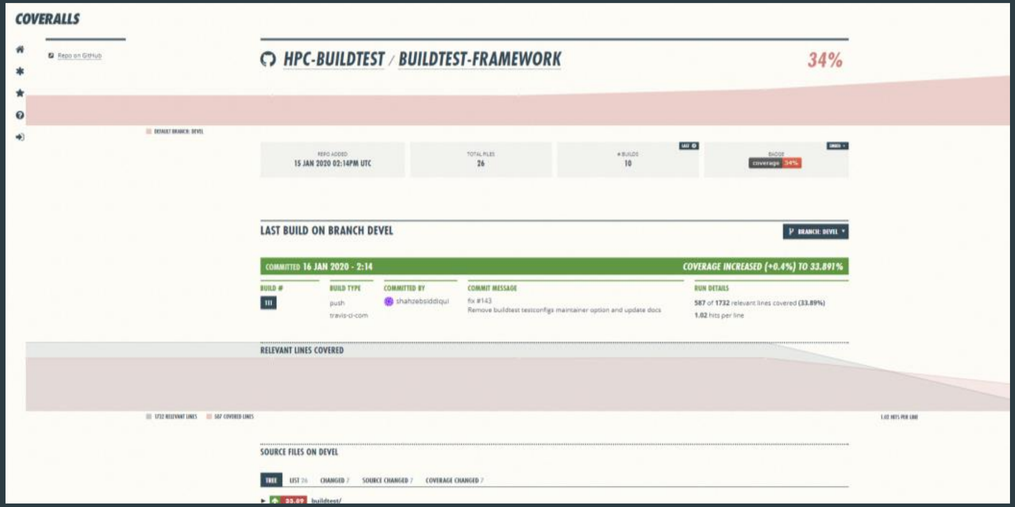
#75 passed
Ran for 4 min 2 sec
Total time 10 min 42 sec
3 days ago

✓	# 75.1	Python: 3.6	LMOD_VERSION=6.6.2	1 min 45 sec
✓	# 75.2	Python: 3.7	LMOD_VERSION=6.6.2	1 min 44 sec
✓	# 75.3	Python: 3.8	LMOD_VERSION=6.6.2	1 min 46 sec
✓	# 75.4	Python: 3.6	LMOD_VERSION=7.8.2	1 min 47 sec
✓	# 75.5	Python: 3.7	LMOD_VERSION=7.8.2	1 min 43 sec
✓	# 75.6	Python: 3.8	LMOD_VERSION=7.8.2	1 min 57 sec

GitHub: https://github.com/HPC-buildtest/buildtest-framework
Documentation: http://buildtest.rtfd.io

Coverage Report

- Since v0.7.5, buildtest can capture coverage report via codecov that is found at https://codecov.io/gh/HPC-buildtest/buildtest-framework
- Codecov report is automatically reported by **codecov** bot on pull requests
- Coveralls provides in-depth and more user-friendly coverage report like codecov



COVERALLS

Repo on GitHub

HPC-BUILDTEST / BUILDTEST-FRAMEWORK 34%

DEFAULT BRANCH: DEVEL

REPO ADDED	TOTAL FILES	# BUILDS	BADGE
15 JAN 2020 02:14PM UTC	26	10	coverage 34%

LAST BUILD ON BRANCH DEVEL

BRANCH: DEVEL

COMMITTED 16 JAN 2020 - 2:14 COVERAGE INCREASED (+0.4%) TO 33.891%

BUILD #	BUILD TYPE	COMMITTED BY	COMMIT MESSAGE	RUN DETAILS
111	push travis-ci-com	shahzebsiddiqui	fix #143 Remove buildtest testconfigs maintainer option and update docs	587 of 1732 relevant lines covered (33.89%) 1.02 hits per line

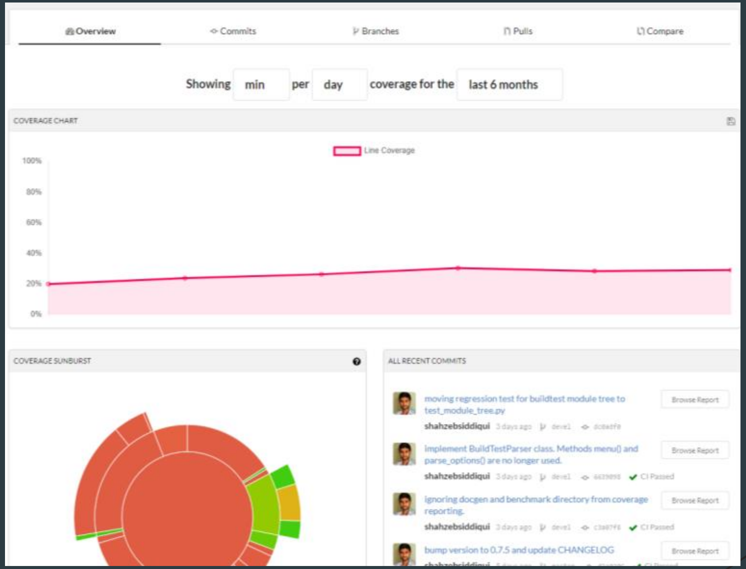
RELEVANT LINES COVERED

1732 RELEVANT LINES 587 COVERED LINES 1.02 HITS PER LINE

SOURCE FILES ON DEVEL

TREE LIST 26 CHANGED 7 SOURCE CHANGED 7 COVERAGE CHANGED 7

33.89 buildtest/



Showing min per day coverage for the last 6 months

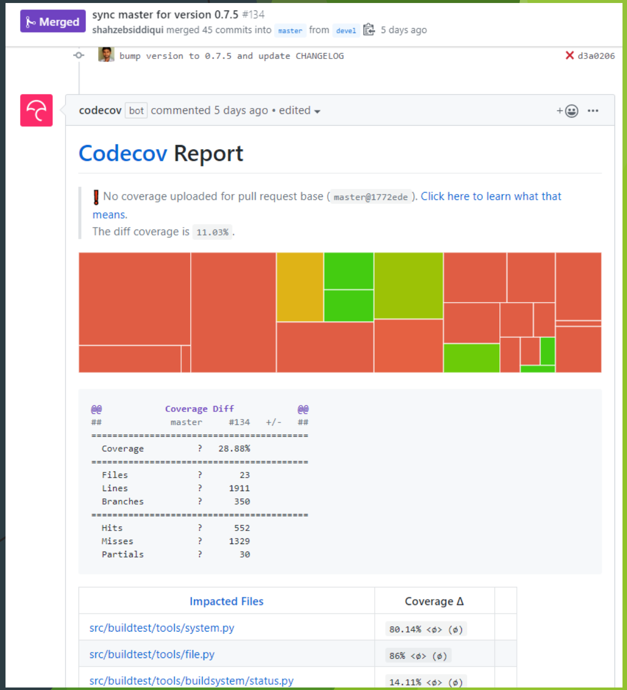
ALL RECENT COMMITS

moving regression test for buildtest module tree to test_module_tree.py
shahzebsiddiqui 3 days ago devel

implement BuildTestParser class. Methods menu() and parse_options() are no longer used.
shahzebsiddiqui 3 days ago devel CI Passed

ignoring docgen and benchmark directory from coverage reporting.
shahzebsiddiqui 3 days ago devel CI Passed

bump version to 0.7.5 and update CHANGELOG



Merged sync master for version 0.7.5 #134
shahzebsiddiqui merged 45 commits into `master` from `devel` 5 days ago

bump version to 0.7.5 and update CHANGELOG ✗ d3a0206

codecov bot commented 5 days ago • edited

Codecov Report

❗ No coverage uploaded for pull request base (`master@1772ede`). Click here to learn what that means.
The diff coverage is `11.03%`.

```
@@           Coverage Diff           @@
##             master     #134   +/-   ##
=========================================
  Coverage          ?   28.88%
=========================================
  Files             ?       23
  Lines             ?     1911
  Branches          ?      350
=========================================
  Hits              ?      552
  Misses            ?     1329
  Partials          ?       30
```

Impacted Files	Coverage Δ
src/buildtest/tools/system.py	80.14% <ø> (ø)
src/buildtest/tools/file.py	86% <ø> (ø)
src/buildtest/tools/buildsystem/status.py	14.11% <ø> (ø)

GitHub Integration

- GitHub Apps Integration
  - CI: Travis
  - Code Quality: CodeCov, Coveralls, CodeFactor
  - Security: Snyk, GuardRails
- GitHub Bot Integration
  - Issue-Label Bot (https://github.com/marketplace/issue-label-bot)
  - Stale (https://github.com/marketplace/stale)
  - Trafico (https://github.com/marketplace/trafico-pull-request-labeler)
  - Pull-Request-Size (https://github.com/marketplace/pull-request-size)
- GitHub Action Integration
  - Black Code Formatter (https://github.com/marketplace/actions/black-code-formatter)
  - URLs-checker (https://github.com/marketplace/actions/urls-checker)

Future Work

- Current YAML schema has some limitation that do not address the following
  - Declaring variables in tests
  - Test permutation (compilation flags, multiple runs, environment variables, compilers)
  - Running test with a range of values (i.e running OpenMP program with range of threads OMP_NUM_THREADS=[1-40])
  - Support for multiple source compilation
- Increase coverage report for regression tests

Reference

Slack Channel	https://hpcbuildtest.slack.com/
Join Slack via Heroku	https://hpcbuildtest.herokuapp.com/
Documentation	http://buildtest.readthedocs.io/
GitHub	https://github.com/HPC-buildtest/buildtest-framework
ReadTheDocs	https://readthedocs.org/projects/buildtest/
Codecov	https://codecov.io/gh/HPC-buildtest/buildtest-framework
Travis	https://travis-ci.com/HPC-buildtest/buildtest-framework
Coverall	https://coveralls.io/github/HPC-buildtest/buildtest-framework
CodeFactor	https://www.codefactor.io/repository/github/hpc-buildtest/buildtest-framework
Snyk	https://app.snyk.io/org/hpc-buildtest/
GuardRails	https://dashboard.guardrails.io/default/gh/HPC-buildtest