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

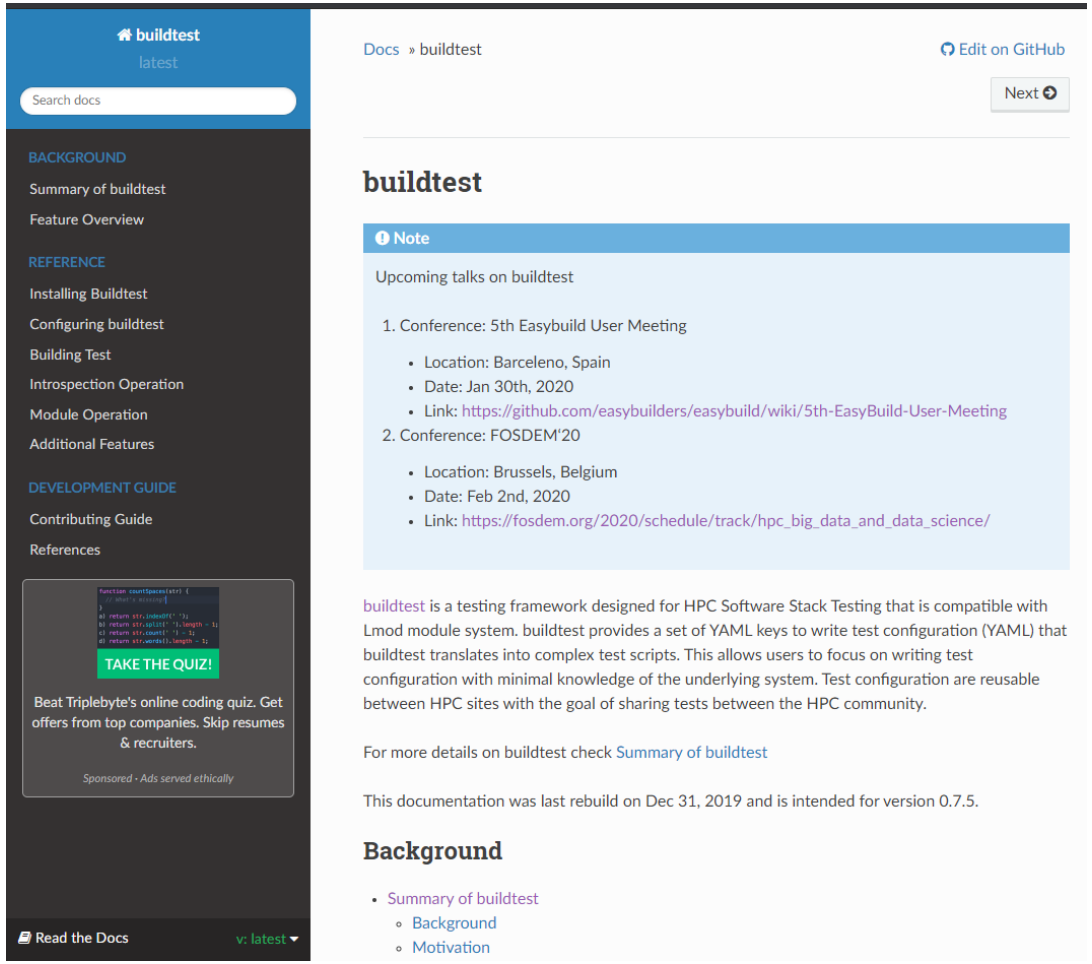
Motivation

- Framework Requirements:
  - The framework is capable of testing of installed software in HPC SW 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 contains a test repository that is community driven
- Buildtest is not meant to replace tools like make, cmake, or autoconf

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

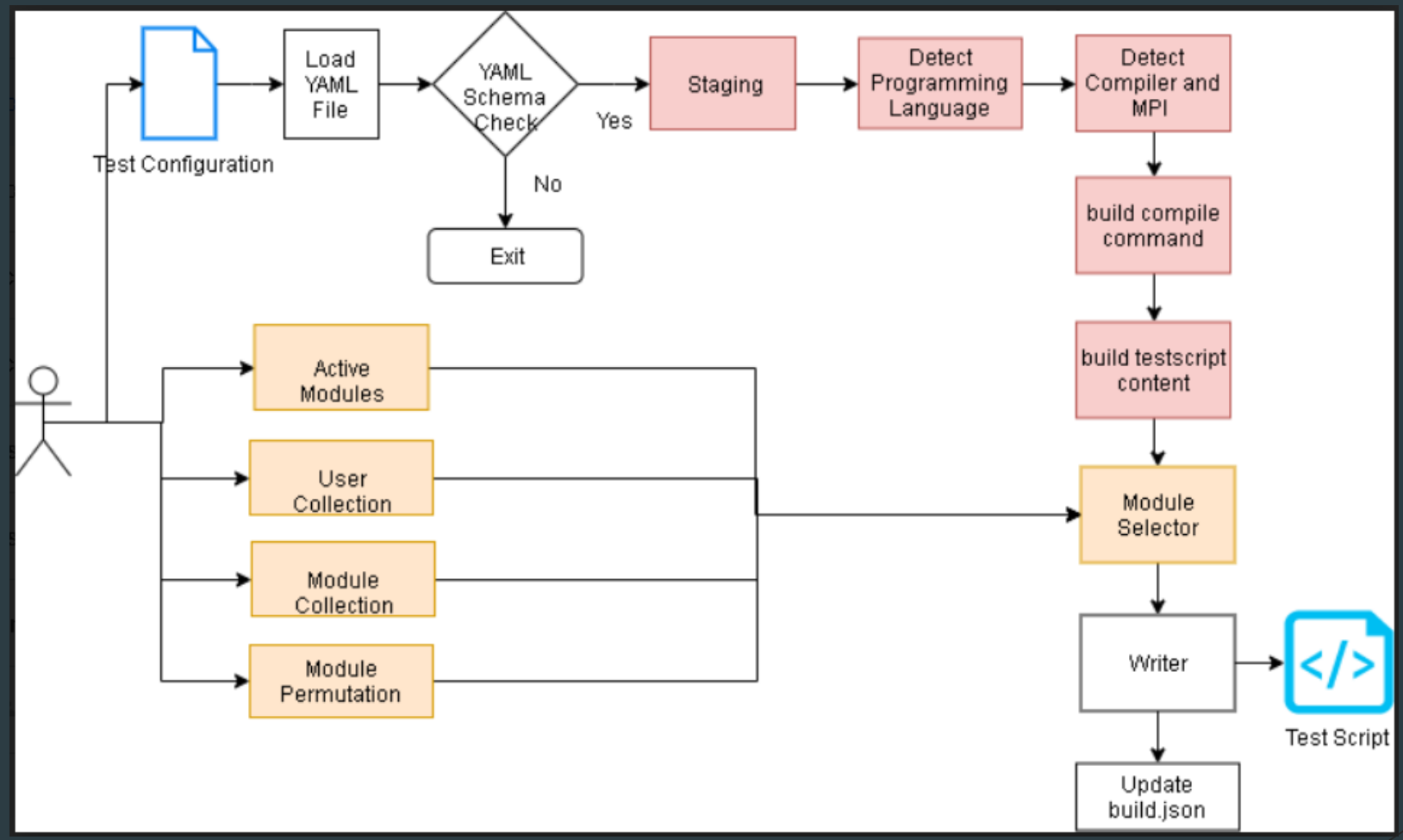
What is buildtest

- 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



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

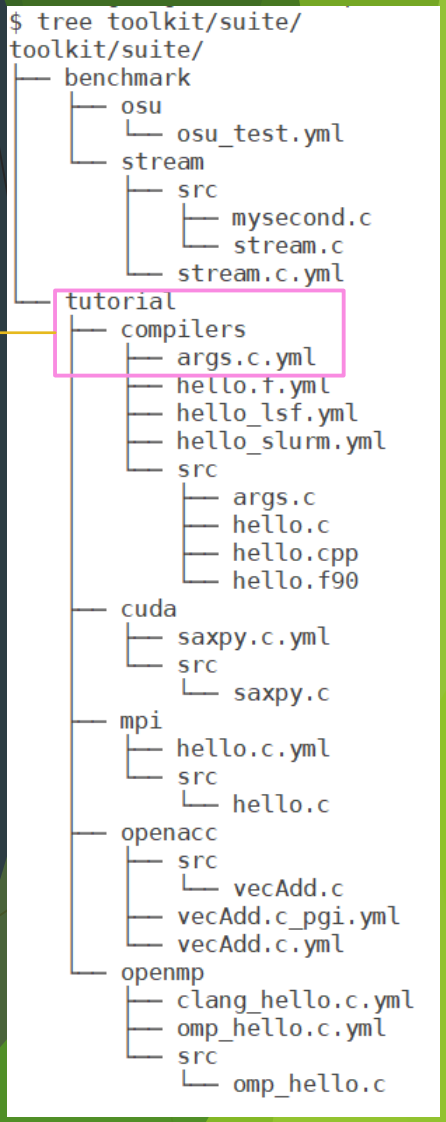
Build Pipeline



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

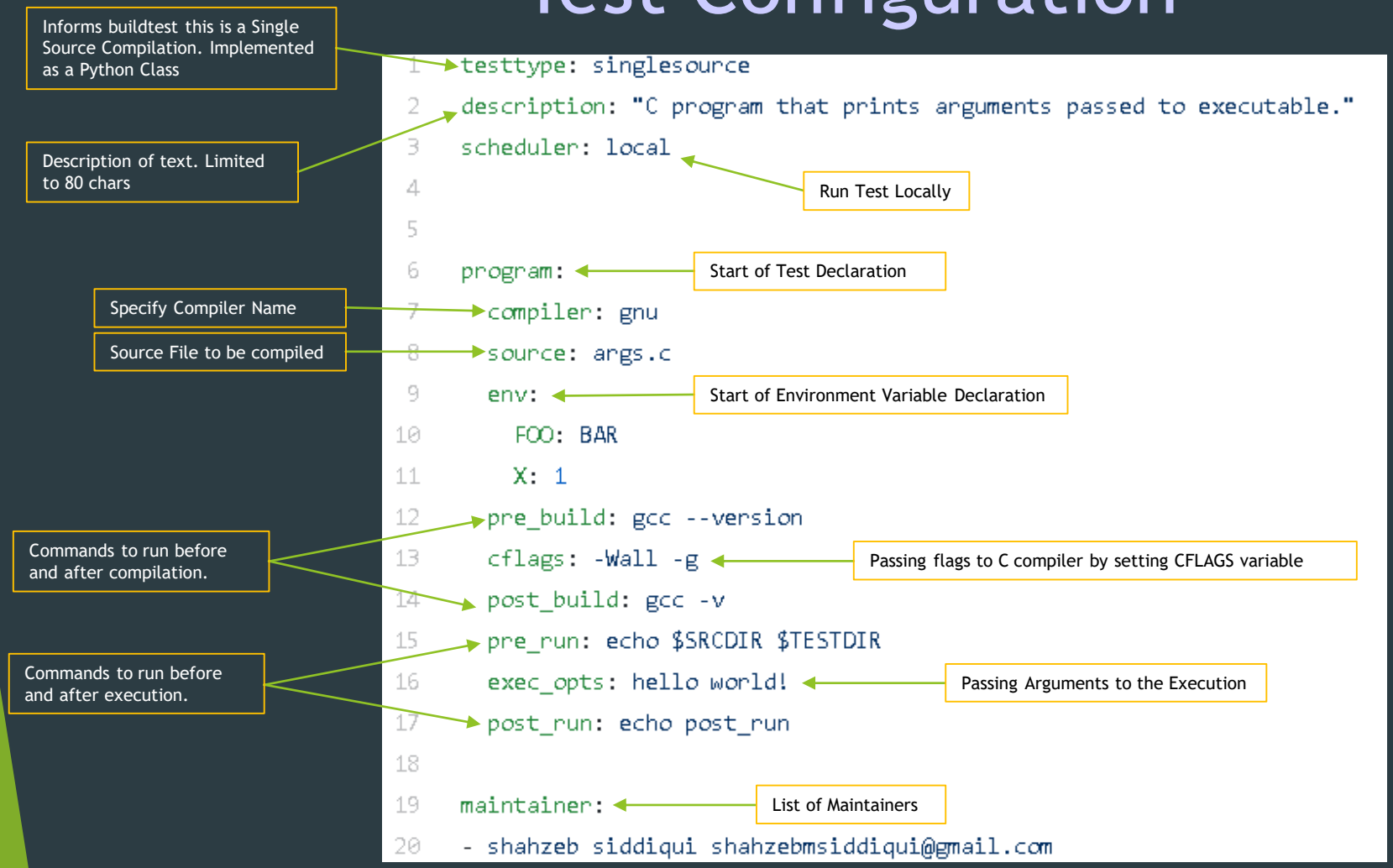
Building a Test

- 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



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

Test Configuration



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

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

_____________________________________________________________________________
```

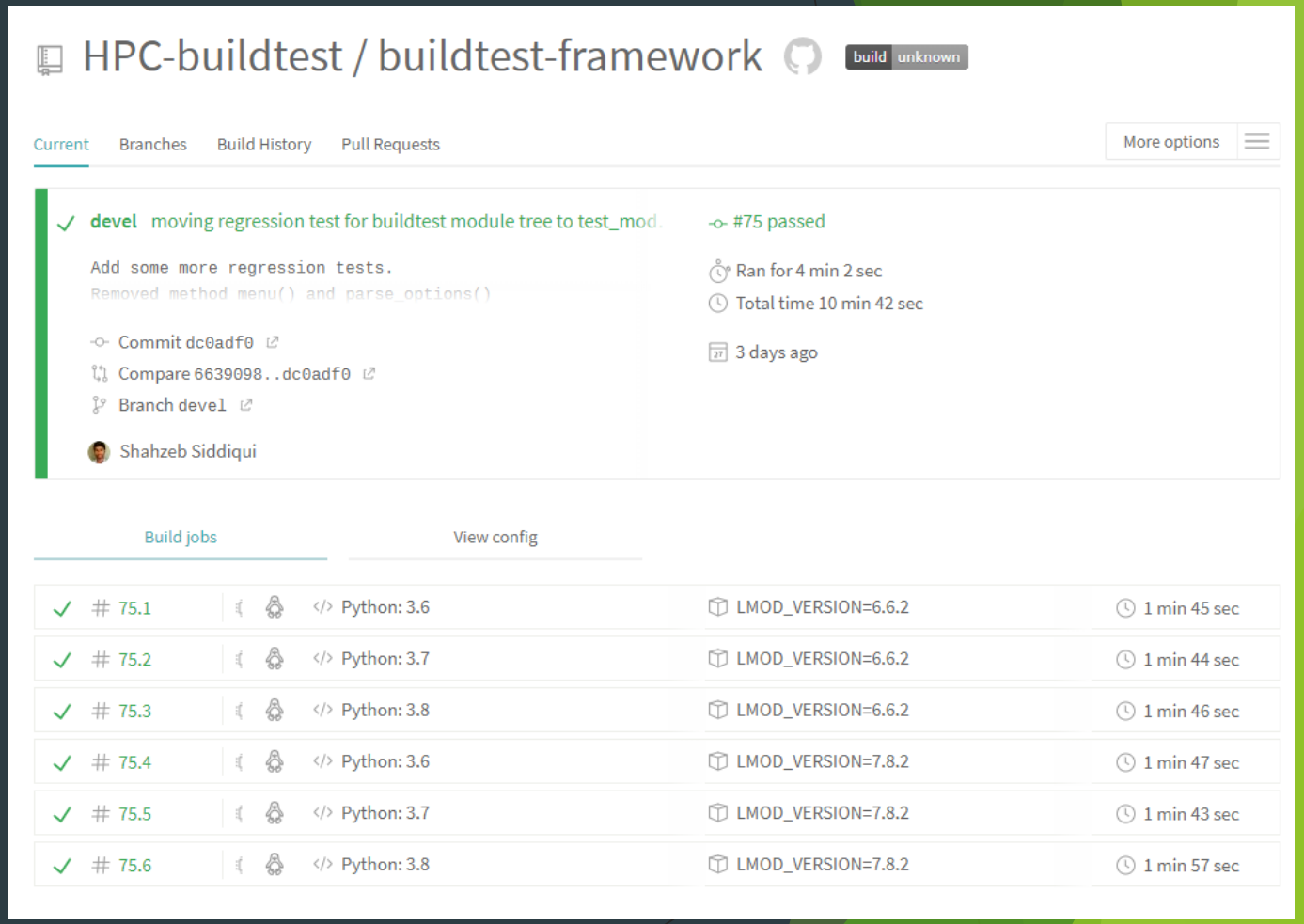
Command Executed

Module File Tested

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

Travis

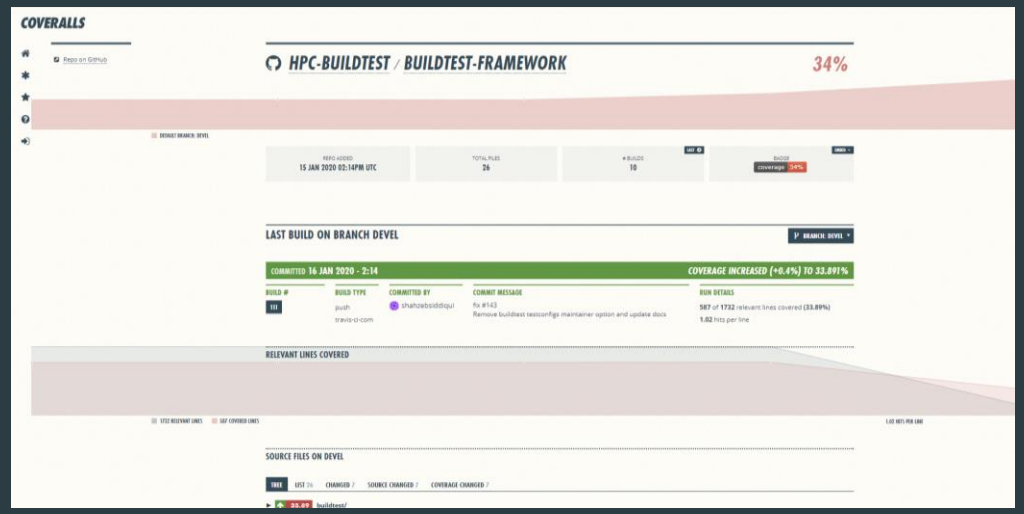
- Since v0.7.4, buildtest can run its regression test in Travis. Several improvement to Travis configuration in v0.7.5
- Currently, buildtest contains approximately 30+ regression tests
- Some regression tests rely on having a software stack, so buildtest builds a mini stack using easybuild.
- Buildtest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

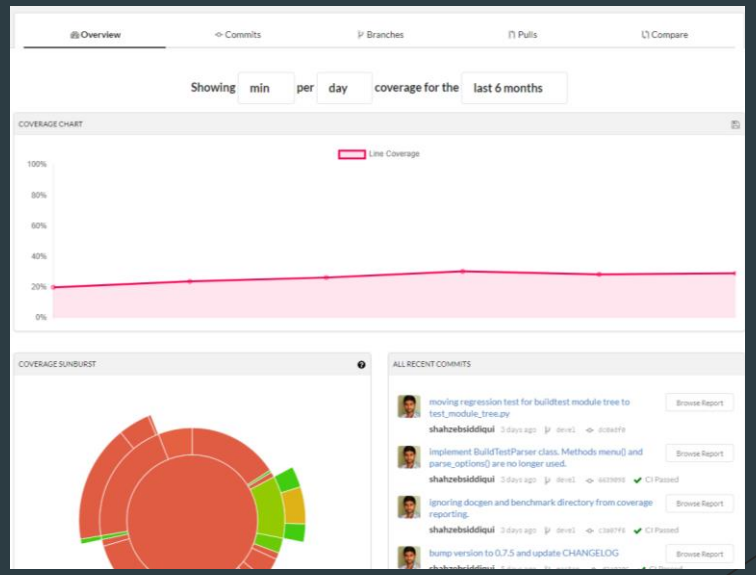


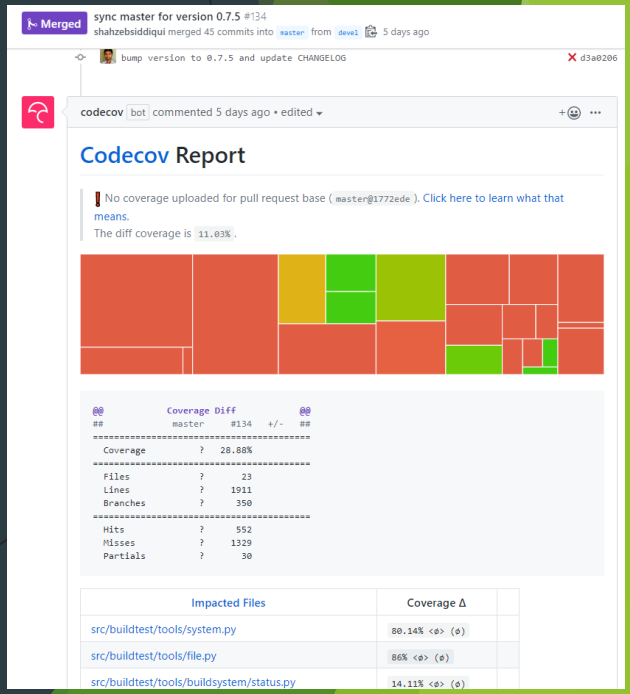
GitHub: https://github.com/HPC-buildtest/buildtest-framework
Documentation: http://buildtest.rtfd.io

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







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

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

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

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

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