



Buildtest: HPC Software Stack Testing Framework

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

Dassault Systemes

5th Easybuild User Meeting

01/30/2020

GitHub: <https://github.com/HPC-buildtest/buildtest-framework>

Documentation: <http://buildtest.rtfd.io>

Motivation

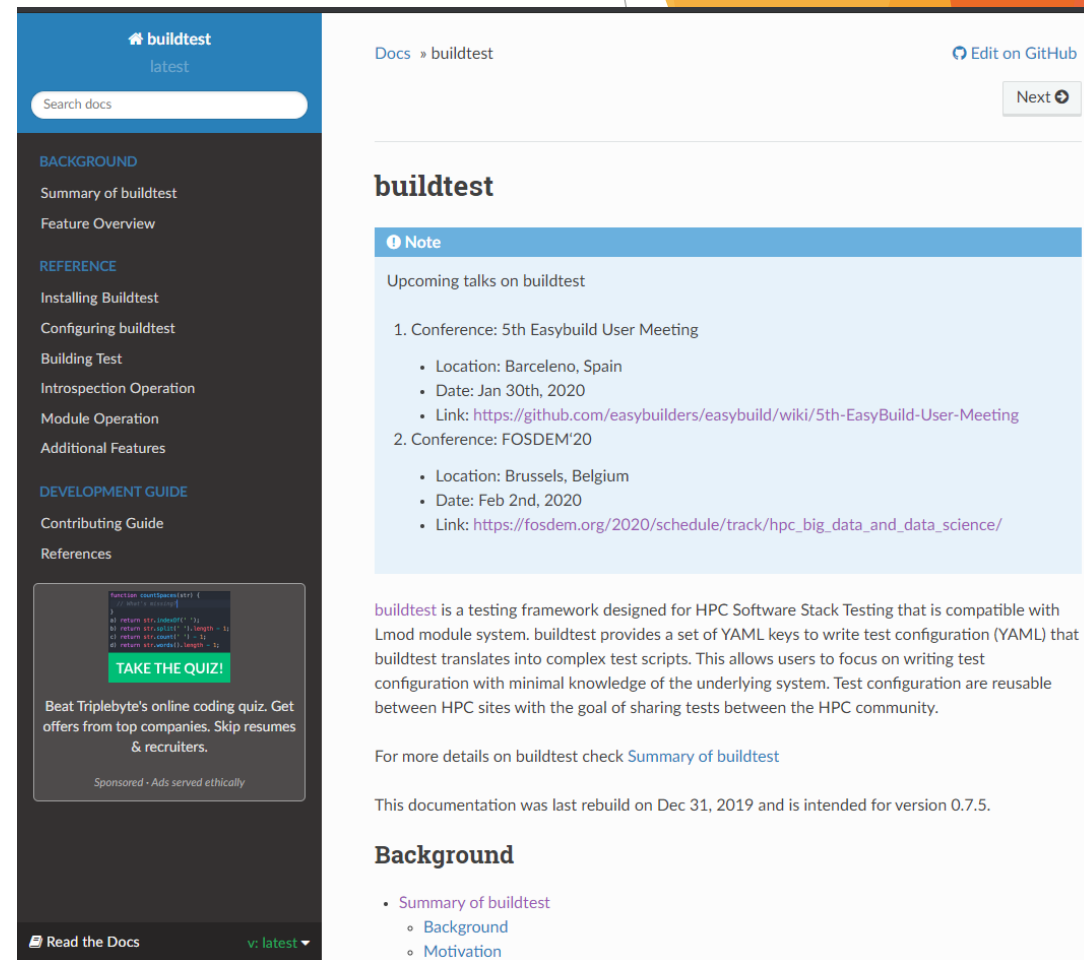
- ▶ Framework Requirements:
 - ▶ The framework is capable of testing of installed software in HPC SW stack
 - ▶ The framework is able to integrate with module system
 - ▶ The framework provides users with a markup language for writing tests
 - ▶ The framework is able to automate test creation and execution
 - ▶ 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.rtf.d.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



The screenshot shows the buildtest documentation website. The left sidebar contains a navigation menu with sections: BACKGROUND (Summary of buildtest, Feature Overview), REFERENCE (Installing Buildtest, Configuring buildtest, Building Test, Introspection Operation, Module Operation, Additional Features), and DEVELOPMENT GUIDE (Contributing Guide, References). The main content area shows the 'buildtest' title, a 'Note' section with upcoming talks, and a description of buildtest as a testing framework for HPC Software Stack Testing. The bottom of the page shows a 'Background' section with links to Summary of buildtest, Background, and Motivation.

buildtest
latest

Search docs

BACKGROUND

- Summary of buildtest
- Feature Overview

REFERENCE

- Installing Buildtest
- Configuring buildtest
- Building Test
- Introspection Operation
- Module Operation
- Additional Features

DEVELOPMENT GUIDE

- Contributing Guide
- References

Take the Quiz!

Beat Triplebyte's online coding quiz. Get offers from top companies. Skip resumes & recruiters.

Sponsored - Ads served ethically

Read the Docs v: latest

Docs » buildtest Edit on GitHub

Next

buildtest

Note

Upcoming talks on buildtest

1. Conference: 5th Easybuild User Meeting
 - Location: Barcelono, 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.

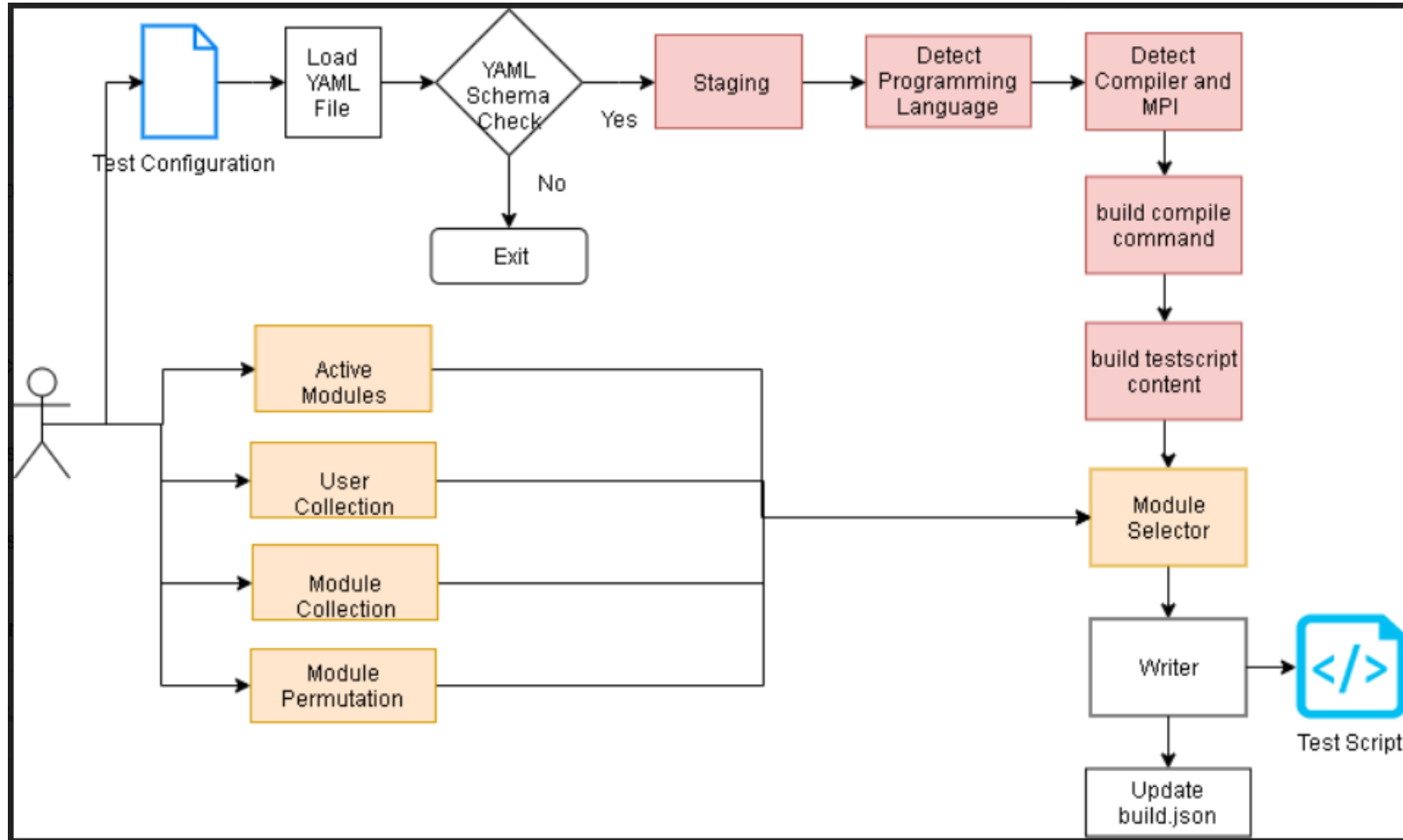
Background

- [Summary of buildtest](#)
 - [Background](#)
 - [Motivation](#)

GitHub: <https://github.com/HPC-buildtest/buildtest-framework>

Documentation: <http://buildtest.rtfid.io>

Build Pipeline



GitHub: <https://github.com/HPC-buildtest/buildtest-framework>
Documentation: <http://buildtest.rtf.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
```

GitHub: <https://github.com/HPC-buildtest/buildtest-framework>
Documentation: <http://buildtest.rtf.d.io>

Test Configuration

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

2 Description of text. Limited to 80 chars

3 Specify Compiler Name

4 Source File to be compiled

5 Commands to run before and after compilation.

6 Commands to run before and after execution.

```
1 testtype: singlesource
2 description: "C program that prints arguments passed to executable."
3 scheduler: local
4
5
6 program:
7 compiler: gnu
8 source: args.c
9 env:
10     FOO: BAR
11     X: 1
12 pre_build: gcc --version
13 cflags: -Wall -g
14 post_build: gcc -v
15 pre_run: echo $SRCDIR $TESTDIR
16 exec_opts: hello world!
17 post_run: echo post_run
18
19 maintainer:
20     - shahzeb siddiqui shahzebmsiddiqui@gmail.com
```

Run Test Locally

Start of Test Declaration

Start of Environment Variable Declaration

Passing flags to C compiler by setting CFLAGS variable

Passing Arguments to the Execution

List of Maintainers

GitHub: <https://github.com/HPC-buildtest/buildtest-framework>
Documentation: <http://buildtest.rtfid.io>

Language, Compiler and MPI Detection

Langage Mapping

Langage	File Extension
c	.c
c++	.cc .cxx .cpp .c++ .C
fortran	.f90 .f95 .f03 .f .F .F90 .FPP .FOR .FTN .for .ftn
cuda	.cu

Compiler Mapping

compiler	language=c	language=c++	language=fortran
gnu	gcc	g++	gfortran
intel	icc	icpc	ifort
pgi	pgcc	pgc++	pgfortran
clang	clang	clang++	N/A
cuda	nvcc	nvcc	N/A

MPI Flavor Mapping

MPI Flavor	language=c	language=c++	language=fortran
openmpi	mpicc	mpicxx	mpif90
intelmapi	mpiicc	mpiicpc	mpiifort
mpich	mpicc	mpicxx	mpif90

GitHub: <https://github.com/HPC-buildtest/buildtest-framework>

Documentation: <http://buildtest.rtf.d.io>

Intel Example

```
1 testtype: singlesource
2 description: Hello World Fortran example using GNU compiler
3 scheduler: local
4
5 program:
6   source: hello.f90
7   compiler: intel
8   fflags: -O2
9
10 maintainer:
11 - shahzeb siddiqui shahzebmsiddiqui@gmail.com
```

```
1 $ buildtest build -c tutorial.compilers.hello.f.yml -co intel --dry
2 Loading Test Configuration (YAML) file: /u/users/ssi29/gpfs/buildtest-framework/toolkit/suite/tutorial/compilers/hello.f.yml
3 Checking schema of YAML file
4 Schema Check Passed
5 Scheduler: local
6 Source Directory: /u/users/ssi29/gpfs/buildtest-framework/toolkit/suite/tutorial/compilers/src
7 Source File: hello.f90
8 Detecting Programming Language, Compiler and MPI wrapper
9 Programming Language: fortran
10 FC: ifort
11 FFLAGS: -O2
12 Test:/tmp/ssi29/buildtest/tests/Intel/Haswell/x86_64/rhel/7.6/build_0/hello.f.yml.0x28f38c1.sh
13 -----
14
15 module purge
16 module restore intel
17 TESTDIR=/tmp/ssi29/buildtest/tests/Intel/Haswell/x86_64/rhel/7.6/build_0
18 SRCDIR=/u/users/ssi29/gpfs/buildtest-framework/toolkit/suite/tutorial/compilers/src
19 SRCFILE=$SRCDIR/hello.f90
20 FC=ifort
21 FFLAGS="-O2"
22 EXECUTABLE=hello.f.yml.0xa7f9d0b4.exe
23
24 cd $TESTDIR
25 $FC $FFLAGS -o $EXECUTABLE $SRCFILE
26 $EXECUTABLE
27 rm ./EXECUTABLE
28 -----
```

GitHub: <https://github.com/HPC-buildtest/buildtest-framework>

Documentation: <http://buildtest.rtf.io>

PGI Example

- ▶ The vecAdd.c program is an OpenACC vector addition program that requires linking to math library (-lm)

```
1 testtype: singlesource
2 scheduler: local
3 description: OpenACC Vector Addition example using GNU compiler
4 maintainer:
5 - shahzeb siddiqui shahzebmsiddiqui@gmail.com
6
7 program:
8   cflags: -acc
9   compiler: pgi
10  ldflags: -lm
11  source: vecAdd.c
```

```
1 $ buildtest build -c tutorial.openacc.vecAdd.c_pgi.yml -co pgi --dry
2 Loading Test Configuration (YAML) file: /u/users/ssi29/gpfs/buildtest-framework/toolkit/suite/tutorial/openacc/vecAdd.c_pgi.yml
3 Checking schema of YAML file
4 Schema Check Passed
5 Scheduler: local
6 Source Directory: /u/users/ssi29/gpfs/buildtest-framework/toolkit/suite/tutorial/openacc/src
7 Source File: vecAdd.c
8 Detecting Programming Language, Compiler and MPI wrapper
9 Programming Language: c
10 CC: pgcc
11 CFLAGS: -acc
12 Test:/tmp/ssi29/buildtest/tests/Intel/Haswell/x86_64/rhel/7.6/build_1/vecAdd.c_pgi.yml.0x20567ea.sh
13 -----
14
15 module purge
16 module restore pgi
17 TESTDIR=/tmp/ssi29/buildtest/tests/Intel/Haswell/x86_64/rhel/7.6/build_1
18 SRCDIR=/u/users/ssi29/gpfs/buildtest-framework/toolkit/suite/tutorial/openacc/src
19 SRCFILE=$SRCDIR/vecAdd.c
20 CC=pgcc
21 CFLAGS="-acc"
22 LDFLAGS="-lm"
23 EXECUTABLE=vecAdd.c_pgi.yml.0x53d8036c.exe
24
25 cd $TESTDIR
26 $CC $CFLAGS -o $EXECUTABLE $SRCFILE $LDFLAGS
27 $EXECUTABLE
28 rm ./$EXECUTABLE
29 -----
```

GitHub: <https://github.com/HPC-buildtest/buildtest-framework>

Documentation: <http://buildtest.rtf.d.io>

Clang Example

- OpenMP example using clang compiler with 2 threads

```
1 testtype: singlesource
2 description: "OpenMP Hello World example in C with clang compiler"
3 scheduler: local
4 program:
5   compiler: clang
6   source: omp_hello.c
7   cflags: -fopenmp
8   pre_exec: OMP_NUM_THREADS=2
9   maintainer:
10  - shahzeb siddiqui shahzebmsiddiqui@gmail.com
```

```
1 $ buildtest build -c tutorial.openmp.clang_hello.c.yml -co Clang --dry
2 Loading Test Configuration (YAML) file: /u/users/ssi29/gpfs/buildtest-framework/toolkit/suite/tutorial/openmp/clang_hello.c.yml
3 Checking schema of YAML file
4 Schema Check Passed
5 Scheduler: local
6 Source Directory: /u/users/ssi29/gpfs/buildtest-framework/toolkit/suite/tutorial/openmp/src
7 Source File: omp_hello.c
8 Detecting Programming Language, Compiler and MPI wrapper
9 Programming Language: c
10 CC: clang
11 CFLAGS: -fopenmp
12 Test:/tmp/ssi29/buildtest/tests/Intel/Haswell/x86_64/rhel/7.6/build_1/clang_hello.c.yml.0x818b7343.sh
13 -----
14
15 module purge
16 module restore Clang
17 TESTDIR=/tmp/ssi29/buildtest/tests/Intel/Haswell/x86_64/rhel/7.6/build_1
18 SRCDIR=/u/users/ssi29/gpfs/buildtest-framework/toolkit/suite/tutorial/openmp/src
19 SRCFILE=$SRCDIR/omp_hello.c
20 CC=clang
21 CFLAGS="-fopenmp"
22 EXECUTABLE=clang_hello.c.yml.0x23a068cc.exe
23
24 cd $TESTDIR
25 $CC $CFLAGS -o $EXECUTABLE $SRCFILE
26 OMP_NUM_THREADS=2 $EXECUTABLE
27 rm ./$EXECUTABLE
28 -----
```

GitHub: <https://github.com/HPC-buildtest/buildtest-framework>

Documentation: <http://buildtest.rtf.d.io>

Build History

- ▶ Buildtest keeps track of every build in a json file ([build.json](#)). The **build ID** that can be used to retrieve tests, logs, and run tests
- ▶ To retrieve a report of all builds: **buildtest build report**

```
$ buildtest build report
```

ID	Build Time	Number of Tests	Command
0	10/20/2019 10:31:30	1	buildtest build -c compilers.helloworld.hello_args.c.yml
1	10/20/2019 10:31:39	8	buildtest build -p gcc
2	10/20/2019 10:31:54	1	buildtest build -c openmp.reduction.omp_reduction.c.yml
3	10/20/2019 10:32:04	5	buildtest build -c openmp.hello.omp_hello.c.yml -m GCC

Running Test Locally

- ▶ Running **buildtest build run <ID>** will run all test scripts that corresponds to the build ID.
- ▶ Buildtest will write a **.run** file that contains output of all tests
- ▶ A zero exit status will be a **PASSED** test and non-zero will be a **FAILED** test

```
$ buildtest build run 2
Running All Tests from Test Directory: /tmp/ssi29/buildtest/tests/Intel/Haswell/x86_64/rhel/7.6/build_2
=====
                        Test summary
Executed 5 tests
Passed Tests: 5 Percentage: 100.0%
Failed Tests: 0 Percentage: 0.0%
SUCCESS: Threshold of 100.0% was achieved
Writing results to /tmp/ssi29/buildtest/tests/Intel/Haswell/x86_64/rhel/7.6/build_2/run/buildtest_09_04_08_11_2019.run
```

GitHub: <https://github.com/HPC-buildtest/buildtest-framework>

Documentation: <http://buildtest.rtfid.io>

Integration with Spider

- ▶ Buildtest solves the module load problem by parsing json content by executing: `spider -o spider-json $BUILDTEST_MODULEPATH`
- ▶ Buildtest leverages spider to load modules into test.
- ▶ Spider is automatically updated when MODULEPATH changes!
- ▶ For more details on spider refer to the documentation:
https://lmod.readthedocs.io/en/latest/136_spider.html

Spider Content

```
"Anaconda3": {  
  "/mxg-hpc/users/ssi29/easybuild/modules/all/Anaconda3/5.3.0.lua": {  
    "Description": "Built to complement the rich, open source Python  
platform \nthat empowers companies to adopt a modern open data science an  
    "URL": "https://www.anaconda.com",  
    "Version": "5.3.0",  
    "fullName": "Anaconda3/5.3.0",  
    "help": "\nDescription\n=====\nBuilt to complement the rich  
ready data analytics platform \nthat empowers companies to adopt a modern  
==\n - Homepage: https://www.anaconda.com\n",  
    "hidden": false,  
    "lpathA": {  
      "/mxg-hpc/users/ssi29/easybuild/software/Anaconda3/5.3.0/lib"  
    },  
    "pV": "000000005.000000003.*zfinal",  
    "pathA": {  
      "/mxg-hpc/users/ssi29/easybuild/software/Anaconda3/5.3.0": 1  
      "/mxg-hpc/users/ssi29/easybuild/software/Anaconda3/5.3.0/bin"  
    },  
    "wV": "000000005.000000003.*zfinal",  
    "whatis": [  
      "Description: Built to complement the rich, open source Python  
s platform \nthat empowers companies to adopt a modern open data science  
      "Homepage: https://www.anaconda.com",  
      "URL: https://www.anaconda.com"  
    ]  
  }  
},
```

GitHub: <https://github.com/HPC-buildtest/buildtest-framework>

Documentation: <http://buildtest.rtf.io>

Reporting Modules

- ▶ Buildtest will retrieve spider record from Lmod and present full canonical module name and path to module file.
- ▶ Buildtest will retrieve modules from all trees defined by **BUILDTEST_MODULEPATH**
- ▶ The default configuration for **buildtest module list** is defined as follows:

```
module:
  list:
    exclude_version_files: true
  filter:
    include: []
  querylimit: -1
```

```
$ buildtest module list
```

Full Module Name	ModuleFile Path
xdusage/2.0	/opt/apps/xsede/modulefiles/xdusage/2.0
xdusage/2.1	/opt/apps/xsede/modulefiles/xdusage/2.1
gateway-usage-reporting/2.0	/opt/apps/xsede/modulefiles/gateway-usage-reporting/2.0
cue-math	/opt/apps/xsede/modulefiles/cue-math
teragrid-basic	/opt/apps/xsede/modulefiles/teragrid-basic
TERAGRID-DEV	/opt/apps/xsede/modulefiles/TERAGRID-DEV
uberftp/2.8	/opt/apps/xsede/modulefiles/uberftp/2.8
cue-comm	/opt/apps/xsede/modulefiles/cue-comm
teragrid-dev	/opt/apps/xsede/modulefiles/teragrid-dev
gsissh/7.1p2	/opt/apps/xsede/modulefiles/gsissh/7.1p2
cue-login-env	/opt/apps/xsede/modulefiles/cue-login-env
globus-6.0	/opt/apps/xsede/modulefiles/globus-6.0
globus/6.0	/opt/apps/xsede/modulefiles/globus/6.0
cue-tg	/opt/apps/xsede/modulefiles/cue-tg
xdinfo/1.3-1	/opt/apps/xsede/modulefiles/xdinfo/1.3-1
TERAGRID-BASIC	/opt/apps/xsede/modulefiles/TERAGRID-BASIC
GLOBUS-6.0	/opt/apps/xsede/modulefiles/GLOBUS-6.0
xdresourceid/1.0	/opt/apps/xsede/modulefiles/xdresourceid/1.0
TERAGRID-paths	/opt/apps/xsede/modulefiles/TERAGRID-paths
cue-build	/opt/apps/xsede/modulefiles/cue-build
CTSSV4	/opt/apps/xsede/modulefiles/CTSSV4
ctssv4	/opt/apps/xsede/modulefiles/ctssv4

```
Total Software Modules: 22
Total LUA Modules: 0
Total non LUA Modules: 22
```

- ▶ **exclude_version_files: true** will ignore **.version**, **.modulerc**, and **.modulerc.lua**
- ▶ **querylimit: -1** Controls number of entries to print. A negative or 0 will print all entries
- ▶ **include: []** filter output by list of module full canonical name

GitHub: <https://github.com/HPC-buildtest/buildtest-framework>

Documentation: <http://buildtest.rtf.d.io>

Reporting Filtered Modules

- ▶ In order to filter modules **GCC, Python, zlib**. You can specify this in buildtest configuration (**settings.yml**) or via command line option **--filter-include**

```
module:
  list:
    filter:
      include: [GCC, Python, zlib]
```

```
1 $ buildtest module list --filter-include "GCC" "Python" "zlib"
2
3 Full Module Name | ModuleFile Path
4 -----|-----
5 GCC/6.4.0-2.28 | /mxg-hpc/users/ssi29/easybuild/modules/all/GCC/6.4.0-2.28.lua
6 GCC/7.1.0-2.28 | /mxg-hpc/users/ssi29/easybuild/modules/all/GCC/7.1.0-2.28.lua
7 GCC/9.2.0-2.32 | /mxg-hpc/users/ssi29/easybuild-HMNS/modules/all/Core/GCC/9.2.0-2.32.lua
8 GCC/8.1.0-2.30 | /mxg-hpc/users/ssi29/easybuild/modules/all/GCC/8.1.0-2.30.lua
9 GCC/8.1.0-2.30 | /mxg-hpc/users/ssi29/easybuild-HMNS/modules/all/Core/GCC/8.1.0-2.30.lua
10 GCC/8.3.0 | /mxg-hpc/users/ssi29/easybuild/modules/all/GCC/8.3.0.lua
11 GCC/7.4.0-2.31.1 | /mxg-hpc/users/ssi29/easybuild/modules/all/GCC/7.4.0-2.31.1.lua
12 Python/3.7.4-GCCcore-8.3.0 | /mxg-hpc/users/ssi29/easybuild/modules/all/Python/3.7.4-GCCcore-8.3.0.lua
13 Python/2.7.14-GCCcore-6.4.0-bare | /mxg-hpc/users/ssi29/easybuild/modules/all/Python/2.7.14-GCCcore-6.4.0-bare.lua
14 zlib/1.2.11 | /mxg-hpc/users/ssi29/easybuild/modules/all/zlib/1.2.11.lua
15 zlib/1.2.11-GCCcore-7.1.0 | /mxg-hpc/users/ssi29/easybuild/modules/all/zlib/1.2.11-GCCcore-7.1.0.lua
16 zlib/1.2.11-GCCcore-8.3.0 | /mxg-hpc/users/ssi29/easybuild/modules/all/zlib/1.2.11-GCCcore-8.3.0.lua
17 zlib/1.2.11 | /mxg-hpc/users/ssi29/easybuild-HMNS/modules/all/Core/zlib/1.2.11.lua
18 zlib/1.2.11-GCCcore-7.4.0 | /mxg-hpc/users/ssi29/easybuild/modules/all/zlib/1.2.11-GCCcore-7.4.0.lua
19 zlib/1.2.11-GCCcore-6.4.0 | /mxg-hpc/users/ssi29/easybuild/modules/all/zlib/1.2.11-GCCcore-6.4.0.lua
20 zlib/1.2.11-zolwez4 | /mxg-hpc/users/ssi29/spack/modules/linux-rhel7-x86_64/Core/zlib/1.2.11-zolwez4.lua
21 zlib/1.2.11-GCCcore-8.1.0 | /mxg-hpc/users/ssi29/easybuild/modules/all/zlib/1.2.11-GCCcore-8.1.0.lua
22
23
24 Total Software Modules: 17
25 Total LUA Modules: 17
26 Total non LUA Modules: 0
```

GitHub: <https://github.com/HPC-buildtest/buildtest-framework>

Documentation: <http://buildtest.rtf.d.io>

Stampede2 Use Case (1 / 3)

- ▶ **Use Case:** User wants to find all modules built for intel/18.

1. User can set root of module tree which will update BUILDTEST_MODULEPATH.

```
(buildtest-framework) login1(997)$ buildtest module tree -s /opt/apps/intel18/modulefiles  
Setting module tree: /opt/apps/intel18/modulefiles  
Configuration File: /home1/06908/sms1990/.buildtest/settings.yml has been updated
```

2. User can view list of module trees

```
(buildtest-framework) login1(998)$ buildtest module tree -l  
/opt/apps/intel18/modulefiles
```

- ▶ Finally user can list the modules via **buildtest module list**

GitHub: <https://github.com/HPC-buildtest/buildtest-framework>

Documentation: <http://buildtest.rtfid.io>

Stampede2 Use Case (2/3)

- ▶ Same output you would see when running **module av**
- ▶ Buildtest will report all modules in tree `/opt/apps/intel/18/modulefiles`

```
(buildtest-framework) login1(999)$ buildtest module list
```

Full Module Name	ModuleFile Path
mvapich2/2.3	/opt/apps/intel18/modulefiles/mvapich2/2.3.lua
mvapich2/2.3.1	/opt/apps/intel18/modulefiles/mvapich2/2.3.1.lua
boost/1.68	/opt/apps/intel18/modulefiles/boost/1.68.lua
boost/1.65	/opt/apps/intel18/modulefiles/boost/1.65.lua
metis/5.0.2	/opt/apps/intel18/modulefiles/mmetis/5.0.2.lua
netcdf/4.6.2	/opt/apps/intel18/modulefiles/netcdf/4.6.2.lua
netcdf/4.3.3.1	/opt/apps/intel18/modulefiles/netcdf/4.3.3.1.lua
basemap/1.1.0	/opt/apps/intel18/modulefiles/basemap/1.1.0.lua
eigen/3.3.4	/opt/apps/intel18/modulefiles/eigen/3.3.4.lua
pybind11/2.2.3	/opt/apps/intel18/modulefiles/pybind11/2.2.3.lua
pdtoolkit/3.25	/opt/apps/intel18/modulefiles/pdtoolkit/3.25.lua
gsl/2.3	/opt/apps/intel18/modulefiles/gsl/2.3.lua
trng/4.21	/opt/apps/intel18/modulefiles/trng/4.21.lua
hdf5/1.8.16	/opt/apps/intel18/modulefiles/hdf5/1.8.16.lua
hdf5/1.10.4	/opt/apps/intel18/modulefiles/hdf5/1.10.4.lua
nco/4.6.9	/opt/apps/intel18/modulefiles/nco/4.6.9.lua
canu/1.8	/opt/apps/intel18/modulefiles/canu/1.8.lua
superlu_seq/5.2.1	/opt/apps/intel18/modulefiles/superlu_seq/5.2.1.lua
silu/4.10.2	/opt/apps/intel18/modulefiles/silu/4.10.2.lua
blis/git20180904	/opt/apps/intel18/modulefiles/blis/git20180904.lua
blis/0.5.0	/opt/apps/intel18/modulefiles/blis/0.5.0.lua
udunits/2.2.25	/opt/apps/intel18/modulefiles/udunits/2.2.25.lua
python3/3.7.0	/opt/apps/intel18/modulefiles/python3/3.7.0.lua
libflame/git20190802	/opt/apps/intel18/modulefiles/libflame/git20190802.lua
swig/3.0.12	/opt/apps/intel18/modulefiles/swig/3.0.12.lua
ncview/2.1.7	/opt/apps/intel18/modulefiles/ncview/2.1.7.lua
loki/0.1.7	/opt/apps/intel18/modulefiles/loki/0.1.7.lua
python2/2.7.15	/opt/apps/intel18/modulefiles/python2/2.7.15.lua
impi/18.0.2	/opt/apps/intel18/modulefiles/impi/18.0.2.lua
impi/18.0.0	/opt/apps/intel18/modulefiles/impi/18.0.0.lua

```
Total Software Modules: 30  
Total LUA Modules: 30  
Total non LUA Modules: 0
```

GitHub: <https://github.com/HPC-buildtest/buildtest-framework>

Documentation: <http://buildtest.rtfid.io>

Stampede2 Use Case (3/3)

- Previously we saw 30 modules, in /opt/app/intel18/modulefiles but there are 320 modules when counting all subtrees.
- Lua and non-lua modules are color coded which can help spot modules quickly

```
(buildtest-framework) login1(1001)$ BUILDTEST_SPIDER_VIEW=all buildtest module list
```

Full Module Name	ModuleFile Path
hpc toolkit/2017.10	/opt/apps/intel18/imp18_0/modulefiles/hpc toolkit/2017.10.lua
parallel-netcdf/4.6.2	/opt/apps/intel18/imp18_0/modulefiles/parallel-netcdf/4.6.2.lua
parallel-netcdf/4.3.3.1	/opt/apps/intel18/imp18_0/modulefiles/parallel-netcdf/4.3.3.1.lua
rosetta/3.9	/opt/apps/intel18/imp18_0/modulefiles/rosetta/3.9.lua
amask/1.0	/opt/apps/intel18/imp18_0/modulefiles/amask/1.0.lua
superlu/5.1-cxxcomplexdebug	/opt/apps/intel18/imp18_0/modulefiles/superlu/5.1-cxxcomplexdebug.lua
superlu/5.1-cxxcomplex	/opt/apps/intel18/imp18_0/modulefiles/superlu/5.1-cxxcomplex.lua
superlu/5.1-cxx	/opt/apps/intel18/imp18_0/modulefiles/superlu/5.1-cxx.lua
superlu/5.1-cxxdebug	/opt/apps/intel18/imp18_0/modulefiles/superlu/5.1-cxxdebug.lua
superlu/5.1-cxxcomplexdebug	/opt/apps/intel18/imp18_0/modulefiles/superlu/5.1-cxxcomplexdebug.lua
superlu/5.1	/opt/apps/intel18/imp18_0/modulefiles/superlu/5.1.lua
superlu/5.1-complex	/opt/apps/intel18/imp18_0/modulefiles/superlu/5.1-complex.lua
superlu/5.1-debug	/opt/apps/intel18/imp18_0/modulefiles/superlu/5.1-debug.lua
pnctcdf/1.11.0	/opt/apps/intel18/imp18_0/modulefiles/pnctcdf/1.11.0.lua
pnctcdf/1.8.1	/opt/apps/intel18/imp18_0/modulefiles/pnctcdf/1.8.1.lua
suitesparse/4.4.3-cxxcomplex	/opt/apps/intel18/imp18_0/modulefiles/suitesparse/4.4.3-cxxcomplex.lua
suitesparse/4.4.3-complex	/opt/apps/intel18/imp18_0/modulefiles/suitesparse/4.4.3-complex.lua
suitesparse/4.4.3-cxxdebug	/opt/apps/intel18/imp18_0/modulefiles/suitesparse/4.4.3-cxxdebug.lua
suitesparse/4.4.3-complexdebug	/opt/apps/intel18/imp18_0/modulefiles/suitesparse/4.4.3-complexdebug.lua
suitesparse/4.4.3-debug	/opt/apps/intel18/imp18_0/modulefiles/suitesparse/4.4.3-debug.lua
suitesparse/4.4.3-cxxcomplexdebug	/opt/apps/intel18/imp18_0/modulefiles/suitesparse/4.4.3-cxxcomplexdebug.lua
suitesparse/4.4.3-cxx	/opt/apps/intel18/imp18_0/modulefiles/suitesparse/4.4.3-cxx.lua
suitesparse/4.4.3	/opt/apps/intel18/imp18_0/modulefiles/suitesparse/4.4.3.lua
nwchem/6.6	/opt/apps/intel18/imp18_0/modulefiles/nwchem/6.6.lua
nwchem/6.8	/opt/apps/intel18/imp18_0/modulefiles/nwchem/6.8.lua
phdf5/1.10.4	/opt/apps/intel18/imp18_0/modulefiles/phdf5/1.10.4.lua
phdf5/1.8.16	/opt/apps/intel18/imp18_0/modulefiles/phdf5/1.8.16.lua

```
hypre/2.14-complex | /opt/apps/intel18/imp18_0/modulefiles/hypre/2.14-complex.lua
meep/1.6 | /opt/apps/intel18/imp18_0/modulefiles/meep/1.6
ls-dyna/11.0.0 | /opt/apps/intel18/imp18_0/modulefiles/ls-dyna/11.0.0.lua
```

```
elemental/0.87-cxxdebug | /opt/apps/intel18/imp18_0/modulefiles/elemental/0.87-cxxdebug.lua
gamsess/2017.04.20 | /opt/apps/intel18/imp18_0/modulefiles/gamsess/2017.04.20.lua
gamsess/2018.02.14 | /opt/apps/intel18/imp18_0/modulefiles/gamsess/2018.02.14.lua
boost-mpi/1.68 | /opt/apps/intel18/imp18_0/modulefiles/boost-mpi/1.68.lua
gpaw/1.5.1 | /opt/apps/intel18/imp18_0/modulefiles/gpaw/1.5.1.lua
hoomd/2.4.2 | /opt/apps/intel18/imp18_0/modulefiles/hoomd/2.4.2.lua
```

```
Total Software Modules: 320
Total LUA Modules: 319
Total non LUA Modules: 1
```

GitHub: <https://github.com/HPC-buildtest/buildtest-framework>

Documentation: <http://buildtest.rtf.d.io>

Module Load Testing (1/3)

- ▶ Buildtest can automate **module load** test for a module tree specified in BUILDTEST_MODULEPATH.
- ▶ Since v0.7.5, buildtest provides a few configuration options to control behavior of module loadtest. The following configuration can be found in **settings.yml**

```
module:  
  loadtest:  
    login: false  
    numtest: -1  
    purge_modules: true
```

- ▶ By default, buildtest will run test in sub-shell but you can run them in **login shell** if **login: true**
- ▶ Buildtest will purge modules before loading modules, this is controlled by **purge_modules: true**
- ▶ User can control how many tests to run before terminating by using **numtest**. A negative or zero will run all tests

GitHub: <https://github.com/HPC-buildtest/buildtest-framework>

Documentation: <http://buildtest.rtfd.io>

Module Load Testing (2/3)

Command Executed

```
$ buildtest module loadtest --numtest 5
RUN: 1 STATUS: PASSED - Testing module command: module purge; module load xdusage/2.0; ( File: /opt/apps/xsede/modulefiles/xdusage/2.0 )
RUN: 2 STATUS: PASSED - Testing module command: module purge; module load xdusage/2.1; ( File: /opt/apps/xsede/modulefiles/xdusage/2.1 )
RUN: 3 STATUS: PASSED - Testing module command: module purge; module load gateway-usage-reporting/2.0; ( File: /opt/apps/xsede/modulefiles/gateway-usage-reporting/2.0 )
RUN: 4 STATUS: FAILED - Testing module command: module purge; module load cue-math; ( File: /opt/apps/xsede/modulefiles/cue-math )
RUN: 5 STATUS: PASSED - Testing module command: module purge; module load teragrid-basic; ( File: /opt/apps/xsede/modulefiles/teragrid-basic )

Writing Results to /tmp/sms1990/buildtest/tests/modules-load.out
Writing Results to /tmp/sms1990/buildtest/tests/modules-load.err

Module Load Summary
Module Trees:                ['/opt/apps/xsede/modulefiles']
PASSED:                      4
FAILED:                      1
```

GitHub: <https://github.com/HPC-buildtest/buildtest-framework>

Documentation: <http://buildtest.rtf.d.io>

Module Load Testing (3/3)

```
$ buildtest module loadtest --login --numtest 5
RUN: 1 STATUS: PASSED - Testing module command: bash --login -c module purge; module load gmpi/2018a; ( File: /mxg-hpc/users/ssi29/easybuild/modules/all/gmpi/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-6.4.0.lua )
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/modules/all/Core', ' /usr/share/lmod/lmod/modulefiles/Core' ]
PASSED: 5
FAILED: 0
```

GitHub: <https://github.com/HPC-buildtest/buildtest-framework>

Documentation: <http://buildtest.rtf.d.io>

Reporting EasyBuild and Spack Modules



```
$ buildtest module --easybuild
Modules built with Easybuild
```

```
-----
/mxg-hpc/users/ssi29/easybuild-HMNS/modules/all/Core/Bison/3.0.4.lua
/mxg-hpc/users/ssi29/easybuild-HMNS/modules/all/Core/Bison/3.3.2.lua
/mxg-hpc/users/ssi29/easybuild-HMNS/modules/all/Core/GCC/8.1.0-2.30.lua
/mxg-hpc/users/ssi29/easybuild-HMNS/modules/all/Core/GCC/9.2.0-2.32.lua
/mxg-hpc/users/ssi29/easybuild-HMNS/modules/all/Core/GCCcore/8.1.0.lua
/mxg-hpc/users/ssi29/easybuild-HMNS/modules/all/Core/GCCcore/9.2.0.lua
/mxg-hpc/users/ssi29/easybuild-HMNS/modules/all/Core/M4/1.4.17.lua
/mxg-hpc/users/ssi29/easybuild-HMNS/modules/all/Core/M4/1.4.18.lua
/mxg-hpc/users/ssi29/easybuild-HMNS/modules/all/Core/binutils/2.30.lua
/mxg-hpc/users/ssi29/easybuild-HMNS/modules/all/Core/binutils/2.32.lua
/mxg-hpc/users/ssi29/easybuild-HMNS/modules/all/Core/flex/2.6.4.lua
/mxg-hpc/users/ssi29/easybuild-HMNS/modules/all/Core/help2man/1.47.4.lua
/mxg-hpc/users/ssi29/easybuild-HMNS/modules/all/Core/zlib/1.2.11.lua
```

```
Total Easybuild Modules: 13
Total Modules Searched: 13
```



```
$ buildtest module --spack
Modules built with Spack
```

```
-----
/mxg-hpc/users/ssi29/spack/modules/linux-rhel7-x86_64/Core/bzip2/1.0.8-etzfbao.lua
/mxg-hpc/users/ssi29/spack/modules/linux-rhel7-x86_64/Core/diffutils/3.7-jthvt3v.lua
/mxg-hpc/users/ssi29/spack/modules/linux-rhel7-x86_64/Core/gdbm/1.18.1-r4vohzu.lua
/mxg-hpc/users/ssi29/spack/modules/linux-rhel7-x86_64/Core/gettext/0.20.1-c4ovdd2.lua
/mxg-hpc/users/ssi29/spack/modules/linux-rhel7-x86_64/Core/libiconv/1.16-xcmzb6a.lua
/mxg-hpc/users/ssi29/spack/modules/linux-rhel7-x86_64/Core/libpciaccess/0.13.5-cavw42z.lua
/mxg-hpc/users/ssi29/spack/modules/linux-rhel7-x86_64/Core/libsigsegv/2.12-oywfthvk.lua
/mxg-hpc/users/ssi29/spack/modules/linux-rhel7-x86_64/Core/libtool/2.4.6-swiq7rt.lua
/mxg-hpc/users/ssi29/spack/modules/linux-rhel7-x86_64/Core/libxml2/2.9.9-azmlgc5.lua
/mxg-hpc/users/ssi29/spack/modules/linux-rhel7-x86_64/Core/m4/1.4.18-dipchcn.lua
/mxg-hpc/users/ssi29/spack/modules/linux-rhel7-x86_64/Core/ncurses/6.1-3jjw2re.lua
/mxg-hpc/users/ssi29/spack/modules/linux-rhel7-x86_64/Core/pkgconf/1.6.3-oqak6dh.lua
/mxg-hpc/users/ssi29/spack/modules/linux-rhel7-x86_64/Core/readline/8.0-bp7xnpf.lua
/mxg-hpc/users/ssi29/spack/modules/linux-rhel7-x86_64/Core/tar/1.32-gem5z6s.lua
/mxg-hpc/users/ssi29/spack/modules/linux-rhel7-x86_64/Core/util-macros/1.19.1-s4xjvop.lua
/mxg-hpc/users/ssi29/spack/modules/linux-rhel7-x86_64/Core/xz/5.2.4-lvajsnj.lua
/mxg-hpc/users/ssi29/spack/modules/linux-rhel7-x86_64/Core/zlib/1.2.11-zolwez4.lua
```

```
Total Spack Modules: 17
Total Modules Searched: 17
```

GitHub: <https://github.com/HPC-buildtest/buildtest-framework>

Documentation: <http://buildtest.rtfid.io>

Retrieving Parent Modules

- ▶ Users and administrators can quickly find all parent modules (modules that set MODULEPATH) with `buildtest module --list-all-parents`.
- ▶ At Stampede2 with up to 900+ modules users can find all parent modules and see all child modules from a given parent using `buildtest module -d <parent-module>`

```
(buildtest-framework) login1(986)$ buildtest module --list-all-parents
gcc/5.4.0 /opt/apps/modulefiles/gcc/5.4.0.lua
gcc/6.3.0 /opt/apps/modulefiles/gcc/6.3.0.lua
gcc/7.1.0 /opt/apps/modulefiles/gcc/7.1.0.lua
impi/17.0.3 /opt/apps/intel17/modulefiles/impi/17.0.3.lua
impi/17.0.3 /opt/apps/intel16/modulefiles/impi/17.0.3.lua
impi/17.0.3 /opt/apps/gcc7_1/modulefiles/impi/17.0.3.lua
impi/17.0.3 /opt/apps/gcc5_4/modulefiles/impi/17.0.3.lua
impi/18.0.0 /opt/apps/gcc7_1/modulefiles/impi/18.0.0.lua
impi/18.0.0 /opt/apps/intel17/modulefiles/impi/18.0.0.lua
impi/18.0.0 /opt/apps/intel18/modulefiles/impi/18.0.0.lua
impi/18.0.0 /opt/apps/gcc6_3/modulefiles/impi/18.0.0.lua
impi/18.0.2 /opt/apps/intel18/modulefiles/impi/18.0.2.lua
impi/18.0.2 /opt/apps/intel17/modulefiles/impi/18.0.2.lua
impi/18.0.2 /opt/apps/gcc6_3/modulefiles/impi/18.0.2.lua
impi/18.0.2 /opt/apps/gcc7_1/modulefiles/impi/18.0.2.lua
intel/16.0.3 /opt/apps/modulefiles/intel/16.0.3.lua
intel/17.0.4 /opt/apps/modulefiles/intel/17.0.4.lua
intel/18.0.0 /opt/apps/modulefiles/intel/18.0.0.lua
intel/18.0.2 /opt/apps/modulefiles/intel/18.0.2.lua
```

```
(buildtest-framework) login1(990)$ buildtest module -d gcc/5.4.0
Module File: /opt/apps/modulefiles/gcc/5.4.0.lua
Modules that depend on gcc/5.4.0
/opt/apps/gcc5_4/modulefiles/impi/17.0.3.lua
/opt/apps/gcc5_4/modulefiles/mkl/17.0.4.lua

Total Modules Found: 2
```

```
(buildtest-framework) login1(995)$ buildtest module -d gcc/6.3.0
Module File: /opt/apps/modulefiles/gcc/6.3.0.lua
Modules that depend on gcc/6.3.0
/opt/apps/gcc6_3/modulefiles/impi/18.0.2.lua
/opt/apps/gcc6_3/modulefiles/impi/18.0.0.lua
/opt/apps/gcc6_3/modulefiles/mkl/18.0.0.lua
/opt/apps/gcc6_3/modulefiles/mkl/18.0.2.lua

Total Modules Found: 4
```

GitHub: <https://github.com/HPC-buildtest/buildtest-framework>

Documentation: <http://buildtest.rtf.d.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

HPG-buildtest / buildtest-framework build unknown

Current Branches Build History Pull Requests More options

✓ **devel** moving regression test for buildtest module tree to test_mod. → #75 passed

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

⌚ Ran for 4 min 2 sec
⌚ Total time 10 min 42 sec
📅 3 days ago

Commit dc0adf0
Compare 6639098...dc0adf0
Branch devel

Shahzeb Siddiqui

Build jobs View config

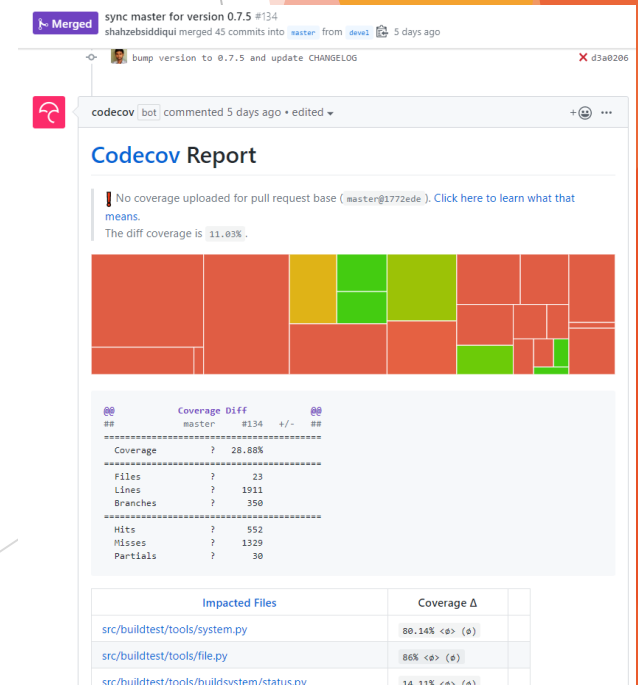
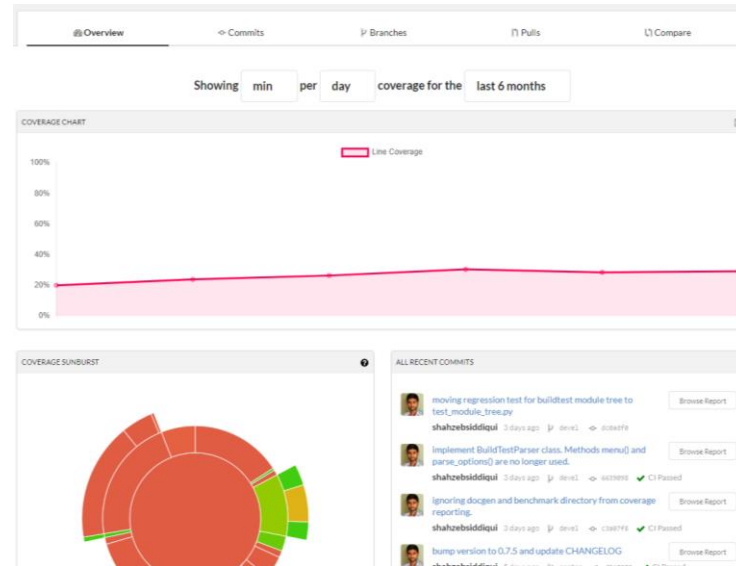
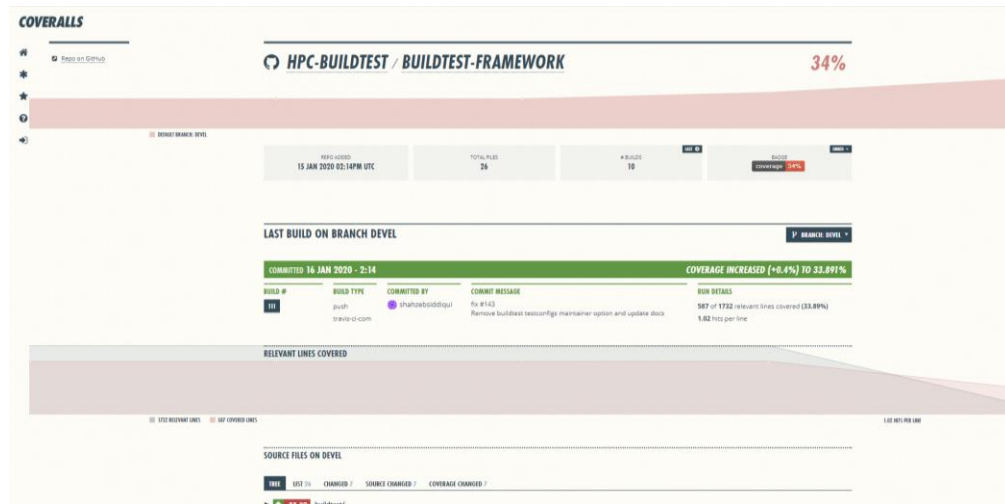
✓ # 75.1		</> Python: 3.6	📦 LMOD_VERSION=6.6.2	⌚ 1 min 45 sec
✓ # 75.2		</> Python: 3.7	📦 LMOD_VERSION=6.6.2	⌚ 1 min 44 sec
✓ # 75.3		</> Python: 3.8	📦 LMOD_VERSION=6.6.2	⌚ 1 min 46 sec
✓ # 75.4		</> Python: 3.6	📦 LMOD_VERSION=7.8.2	⌚ 1 min 47 sec
✓ # 75.5		</> Python: 3.7	📦 LMOD_VERSION=7.8.2	⌚ 1 min 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.rtfid.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>
- ▶ Coverage 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.rtfid.io>

Current Challenges

- ▶ Travis Build Timeouts if job takes more than 50mins. Buildtest requires a software stack to run some of its regression tests
- ▶ Currently, buildtest installs easybuild and lets eb install 2-3 software to get software stack to test some module functionality
- ▶ One approach is to build easybuild containers via singularity and pull images in Travis build and rewrite some regression test to test against containers

GitHub: <https://github.com/HPC-buildtest/buildtest-framework>

Documentation: <http://buildtest.rtdf.io>

Easybuild Singularity Container Workflow

```
#1. Generate Singularity Definition File from Easybuild
eb binutils-2.28.eb -C --experimental --container-config bootstrap=yum,osversion=7

#2. Create Access Token from Sylabs and login
singularity remote login

#3. build container on Sylabs builder
singularity build --remote binutils-2.28.sif Singularity.binutils-2.28

#4a. Generate Key if you dont have one. If you have key then sign the container
singularity key newpair

#4b. List your Key pair
singularity key list

#4c. In Step 4a. if you select N in Would you like to push it to the keystore? [Y,n]
singularity key push <KEY>

#5. Sign & Verify your container
singularity sign binutils-2.28.sif
singularity verify binutils-2.28.sif

#6. Push container to your library
singularity push binutils-2.28.sif library://shahzebmsiddiqui/easybuild/binutils:2.28
```

GitHub: <https://github.com/HPC-buildtest/buildtest-framework>

Documentation: <http://buildtest.rtfid.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>)

GitHub: <https://github.com/HPC-buildtest/buildtest-framework>

Documentation: <http://buildtest.rtf.d.io>

Autoformat Python Code with Black

- Buildtest codebase is automatically formatted upon **push** and **pull request** using GitHub Workflow Action **Black Code Formatter** (<https://github.com/marketplace/actions/black-code-formatter>)

The screenshot displays the GitHub repository interface for `HPC-buildtest / buildtest-framework`. The top navigation bar includes links for Code, Issues (5), Pull requests (0), Actions, Projects (0), Wiki, Security, Insights, and Settings. On the right, there are buttons for Sponsor, Used by (5), Unwatch (4), Unstar (25), and Fork (4).

The main content area shows a workflow run titled `update documentation` on the `devel` branch, triggered by a push (commit `67a788a`). The workflow is currently running, and the `black-format` job is highlighted with a green checkmark.

The `Black Formatter / black-format` job log shows the following steps and output:

- Set up job (3s)
- Build lgeiger/black-action@master (46s)
- Run actions/checkout@v1 (2s)
- Black Code Formatter (3s)

The output of the Black Code Formatter job is as follows:

```
1 ▶ Run lgeiger/black-action@master
4 /usr/bin/docker run --name ee63b13d81dbe344d17b1f5cb684953a3a6_37a27c --label 671ee6 --workdir /github/workspace --rm -e INPUT_ARGS -e HOME -e GITHUB_REF -e GITHUB_SHA -e GITHUB_REPOSITORY -e GITHUB_ACTOR -e GITHUB_WORKFLOW -e GITHUB_HEAD_REF -e GITHUB_BASE_REF -e GITHUB_EVENT_NAME -e GITHUB_WORKSPACE -e GITHUB_ACTION -e GITHUB_EVENT_PATH -e RUNNER_OS -e RUNNER_TOOL_CACHE -e RUNNER_TEMP -e RUNNER_WORKSPACE -e ACTIONS_RUNTIME_URL -e ACTIONS_RUNTIME_TOKEN -e GITHUB_ACTIONS=true -v "/var/run/docker.sock":"/var/run/docker.sock" -v "/home/runner/work/_temp/_github_home":"/github/home" -v "/home/runner/work/_temp/_github_workflow":"/github/workflow" -v "/home/runner/work/buildtest-framework/buildtest-framework":"/github/workspace" 671ee6:3b13d81dbe344d17b1f5cb684953a3a6 .
5 reformatted /github/workspace/setup.py
6 reformatted /github/workspace/docs/conf.py
7 reformatted /github/workspace/src/buildtest/docgen/main.py
8 reformatted /github/workspace/src/buildtest/tools/config.py
9 reformatted /github/workspace/src/buildtest/tools/list.py
10 reformatted /github/workspace/src/buildtest/tools/buildsystem/singlesource.py
11 reformatted /github/workspace/src/buildtest/tools/modules.py
12 reformatted /github/workspace/src/buildtest/tools/writer.py
13 reformatted /github/workspace/tests/test_file.py
14 reformatted /github/workspace/tests/test_inspect.py
15 reformatted /github/workspace/tests/test_module_collection.py
16 reformatted /github/workspace/src/buildtest/tools/system.py
17 reformatted /github/workspace/tests/test_modules.py
18 All done! 🎉 📦 🎉
19 13 files reformatted, 31 files left unchanged.
```

The job concludes with a green checkmark and the text `Complete job` (0s).

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
- ▶ Merge building and execution of test into one operation. Currently the two operations are separated
- ▶ Increase coverage report for regression tests

GitHub: <https://github.com/HPC-buildtest/buildtest-framework>

Documentation: <http://buildtest.rtf.d.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.rtf.d.io>