Facility Testing of E4S via E4S Testsuite, Spack Test, and buildtest

Approved for public release

Shahzeb Siddiqui (Lawrence Berkeley National Laboratory)

Sep 14th 2021

ECP Event: https://www.exascaleproject.org/event/buildtest-21-09/

https://buildtest.readthedocs.io/

https://github.com/buildtesters/buildtest

http://hpcbuildtest.slack.com/

National Nuclear Security Administration

U.S. DEPARTMENT OF ENERGY | Office of Science

About Me

- I am an HPC Consultant at NERSC in the **User Engagement Group** that is responsible for user support including support tickets, user documentation, training, and managing software stack for NERSC.
- I am the L4 for Software Integration Group (WBS: 2.4.4.01) in the ECP Project. In this group we are responsible for deploying the Extreme Scale Scientific Software Stack (E4S) at the DOE Facilities (NERSC, OLCF, ALCF)
- Previously held multiple roles throughout my career including Dassault Systems, Pfizer, Penn State, IBM, NASA, and Northrop Grumman
- Creator of buildtest: HPC Testing Framework
- Certified Red Hat Certified System Administrator (RHCSA): 200-019-677
- Masters in Computer Science from KAUST

KAUST
King Abdullah University of Science and Technology

https://www.linkedin.com/in/shahzebmsiddiqui/

https://github.com/shahzebsiddiqui/

E4S

Container Releases

Docker Download

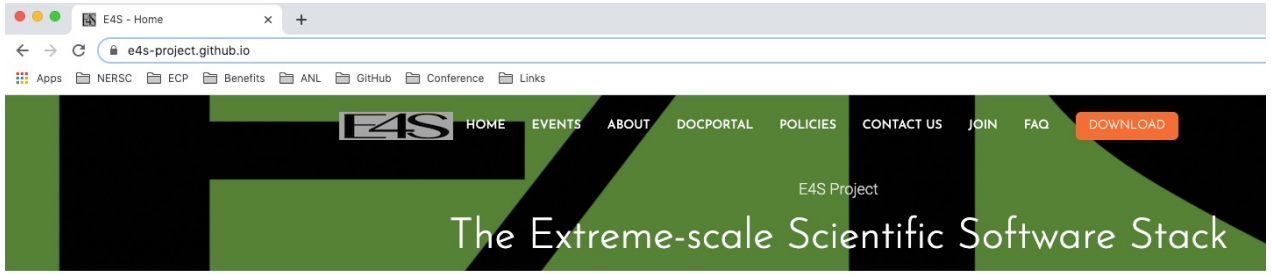
Singularity x86_64 Download

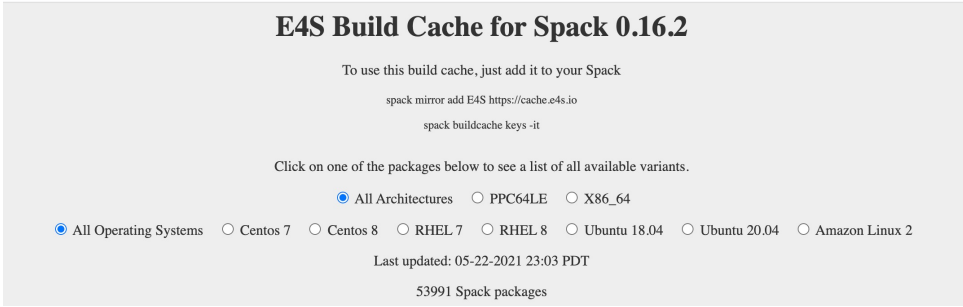
Singularity ppc64le Download

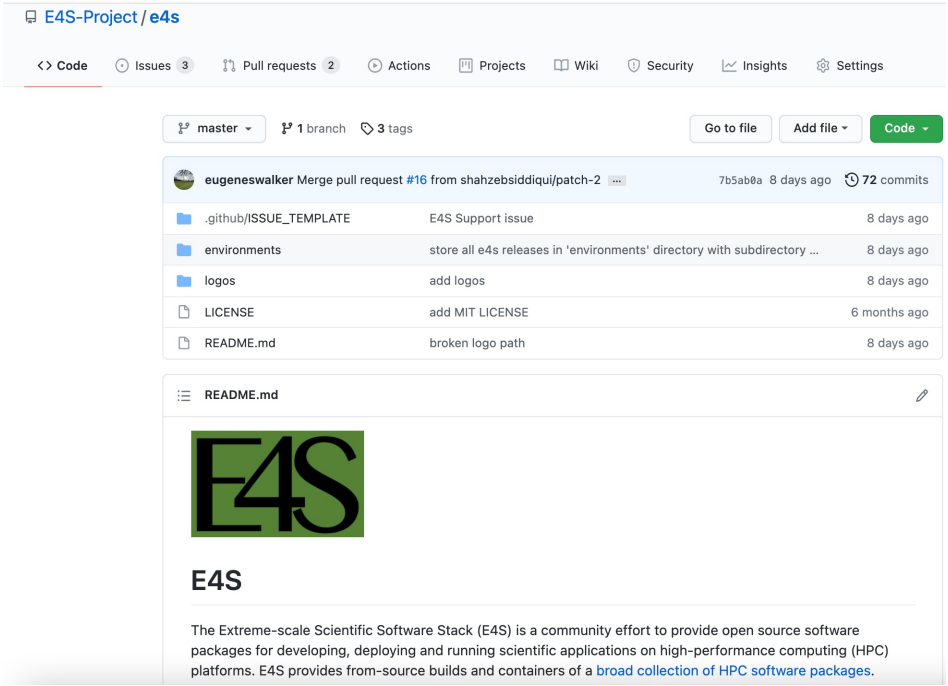
CharlieCloud Download

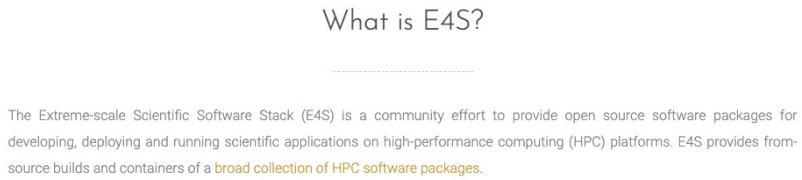
OVA Download

- Extreme-scale Scientific Software Stack (E4S) is a curated set of software packages for developing and running scientific application on HPC platforms.
- E4S is a subset of Spack Packages
- E4S is deployed as spack manifest, containers, and buildcache.
- Contains up to 80+ software products including: compilers, data and viz tools, I/O tools, profilers, xSDK and may more









E4S: Extreme-scale Scientific Software Stack

- Curated, Spack based software distribution
- Spack binary build caches for bare-metal installs
  - x86_64, ppc64le (IBM Power 9), and aarch64 (ARM64)
- Container images on DockerHub and E4S website of pre-built binaries of ECP ST products
- Base images and full featured containers (with GPU support)
- GitHub recipes for creating custom images from base images
- GitLab integration for building E4S images
- E4S validation test suite on GitHub
- E4S-cl container launcher tool for MPI substitution in applications using MPICH ABI
- E4S VirtualBox image with support for container runtimes
  - Docker
  - Singularity
  - Shifter
  - Charliecloud
- AWS and GCP images to deploy E4S

https://e4s.io

https://e4s.io/talks/E4S_IAW21.pptx (Slide 41)

Extreme-scale Scientific Software Stack (E4S)

- E4S: HPC Software Ecosystem – a curated software portfolio
- A **Spack-based** distribution of software tested for interoperability and portability to multiple architectures
- Available from **source**, **containers**, **cloud, binary caches**
- Leverages and enhances SDK interoperability thrust
- Not a commercial product – an open resource for all
- Oct 2018: E4S 0.1 - 24 full, 24 partial release products
- Jan 2019: E4S 0.2 - 37 full, 10 partial release products
- Nov 2019: E4S 1.0 - 50 full, 5 partial release products
- Feb 2020: E4S 1.1 - 61 full release products
- Nov 2020: E4S 1.2 (aka, 20.10) - 67 full release products
- Feb 2021: E4S 21.02 - 67 full release, 4 partial release
- May 2021: E4S 21.05 - 76 full release products

E4S

https://e4s.io

Lead: Sameer Shende
(U Oregon)

Also include other products .e.g.,
AI: PyTorch, TensorFlow, Horovod
Co-Design: AMReX, Cabana, MFEM

https://e4s.io/talks/E4S_IAW21.pptx (Slide 44)

E4S Facility Deployment Dashboard

1.2. E4S Facility Deployment Dashboard

Cori

E4S Version	Project URL (Github, Gitlab)	Installed Specs	Compiler	Spack commit	Spack.yaml	Installed Specs
20.10	https://software.nersc.gov/NERSC/e4s-2010	135	intel@19.1.2.254	e1e0bbb4cbe11a3f0d7e50466ffa86071ee653b7	https://github.com/spack/spack-configs/blob/master/NERSC/cori/e4s-20.10/spack.yaml	https://github.com/spack/spack-configs/blob/master/NERSC/cori/e4s-20.10/e4s-20.10.txt
21.02	https://software.nersc.gov/NERSC/e4s-2102	149	intel@19.1.2.254 and gcc@10.1.0	b56d65fce5f4743a23399f0cde006bed1b52d53d	https://github.com/spack/spack-configs/blob/main/NERSC/cori/e4s-21.02/spack.yaml	https://github.com/spack/spack-configs/blob/main/NERSC/cori/e4s-21.02/e4s-21.02.txt
21.05	https://github.com/spack/spack-configs	157	intel@19.1.3.304	https://github.com/spack/spack/tree/e4s-21.05	https://github.com/spack/spack-configs/blob/main/NERSC/cori/e4s-21.05/spack.yaml	https://github.com/spack/spack-configs/blob/main/NERSC/cori/e4s-21.05/e4s-21.05.txt

Perlmutter

E4S Version	Project URL (Github, Gitlab)	Installed Specs	Compiler	Spack commit	Spack.yaml	Installed Specs

Summit

E4S Version	Project URL (Github, Gitlab)	Installed Specs	Compiler	Spack commit	Spack.yaml	Installed Specs
20.10		48/70	gcc@6.4.0	N/A	N/A	https://docs.olcf.ornl.gov/software/e4s.html

Spock

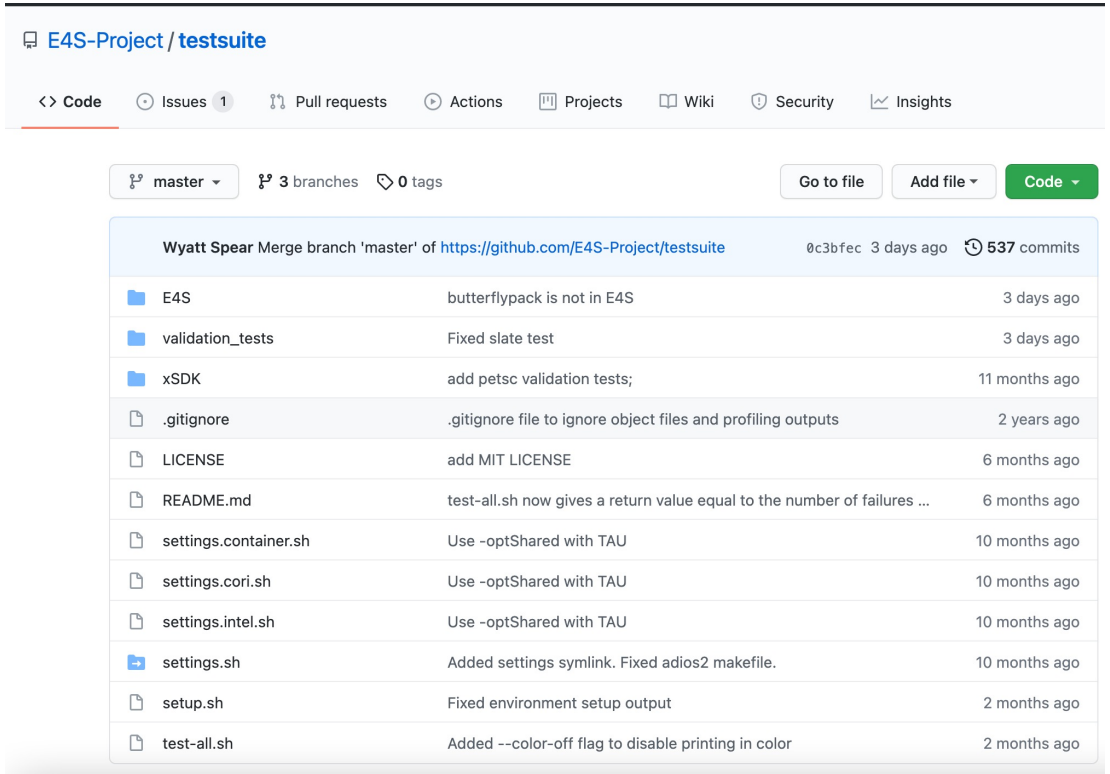
E4S Version	Project URL (Github, Gitlab)	Installed Specs	Compiler	Spack commit	Spack.yaml	Installed Specs
21.02		40/75	gcc@10.2.0 and others	N/A	N/A	https://docs.olcf.ornl.gov/software/e4s.html#spock

Articus

E4S Version	Project URL (Github, Gitlab)	Installed Specs	Compiler	Spack commit	Spack.yaml	Installed Specs
21.05	**https://github.com/frankwillmore/deployment**	334	gcc@9.3.0	https://github.com/spack/spack/tree/e4s-21.05	https://github.com/spack/spack-configs/blob/main/ANL/JLSE/Arcticus/E4S-21.05/spack.yaml	https://github.com/spack/spack-configs/blob/main/ANL/JLSE/Arcticus/E4S-21.05/e4s-21.05.txt

E4S Test Suite

- The E4S Test Suite is a collection of tests to validate E4S stack and increase test coverage for deployed stack.
- The main script test-all.sh can be run as standalone program which will test everything or you can specify an argument to a directory of tests to run.



E4S Testsuite on Cori

```
siddiq90@cori04> cat testsuite.sh
#!/bin/bash
git clone https://github.com/E4S-Project/testsuite.git
cd testsuite
module load e4s/21.05
source ./setup.sh
./test-all.sh validation_tests/gasnet --settings settings.cori.sh
```

```
siddiq90@cori04> sh testsuite.sh
Cloning into 'testsuite'...
remote: Enumerating objects: 18242, done.
remote: Counting objects: 100% (843/843), done.
remote: Compressing objects: 100% (523/523), done.
remote: Total 18242 (delta 494), reused 576 (delta 307), pack-reused 17399
Receiving objects: 100% (18242/18242), 35.07 MiB | 20.45 MiB/s, done.
Resolving deltas: 100% (8858/8858), done.
Updating files: 100% (14090/14090), done.
===
validation_tests/gasnet
sbw7ukx
Cleaning /global/homes/s/siddiq90/testsuite/validation_tests/gasnet
Compiling /global/homes/s/siddiq90/testsuite/validation_tests/gasnet
Running /global/homes/s/siddiq90/testsuite/validation_tests/gasnet
Success
```

```
siddiq90@cori04> spack find --format "{hash} {name}@{version}%{compiler}" gasnet
sbw7ukx5qjxonvp6oj62qxorzdve3lid gasnet@2021.3.0%intel@19.1.3.304
```

spack test: write tests directly in Spack packages, so that they can evolve with the software

```
class Libsigsegv(AutotoolsPackage, GNUMirrorPackage):
    """GNU libsigsegv is a library for handling page faults in user mode."""

    # ... spack package contents ...

    extra_install_tests = ‘tests/.libs’

    def test(self):
        data_dir = self.test_suite.current_test_data_dir
        smoke_test_c = data_dir.join(‘smoke_test.c’)

        self.run_test(
            'cc’, [
                '-I%s' % self.prefix.include,
                '-L%s' % self.prefix.lib, '-lsigsegv’,
                smoke_test_c,
                '-o', 'smoke_test'
            ]
            purpose='check linking’)

        self.run_test(
            ‘smoke_test’, [], data_dir.join('smoke_test.out’),
            purpose=‘run built smoke test’)

        self.run_test('sigsegv1': ['Test passed’], purpose='check sigsegv1 output’)
        self.run_test('sigsegv2': ['Test passed’], purpose='check sigsegv2 output’)
```

Tests are part of a regular Spack recipe class

Easily save source code from the package

User just defines a `test()` method

Retrieve saved source.
Link a simple executable.

Spack ensures that `CC` is a compatible compiler

Run the built smoke test and verify output

Run programs installed with package

Spack Test Command Line Overview

Command	Description
spack test list	List tests for installed packages
spack test list --all	List all tests for all spack packages
spack test run	Run test for all installed specs in environment or installed packages
spack test run hdf5	Run test for spack package hdf5
spack test run --alias hdf5 hdf5@1.10.7	Run test for hdf5@1.10.7 and assign alias for suite name **hdf5**
spack test results	Show results for all test suites
spack test results – hdf5@1.10.7	Show test results for spec hdf5@1.10.7
spack test results <suite-name>	Show test results for suite name
spack test remove -y	Remove all test results and assume ‘yes’ for each confirmation

```
siddiq90@cori04> spack test list
-- cray-cnl7-haswell / intel@19.1.3.304 -------------------------
u422bud arborx@1.0        j7o7wnp hypre@2.20.0       wuamopr parallel-netcdf@1.12.2    ypjok7i py-psutil@5.7.2        ownfza6 py-warpx@21.05    fmkmiba superlu@5.2.1
vd2zjhd binutils@2.33.1   z5kxptz kokkos@3.4.00      cz6lxgn py-libensemble@0.7.2      edkcfue py-pyelftools@0.26     3iyxhqw python@3.8.10     2rrh7bg superlu-dist@6.4.0
opjamiu cmake@3.20.2      hl26wxt kokkos@3.4.00      fxlw6ij py-mpi4py@3.0.3           sjmnbud py-pyparsing@2.4.7     g5h4vwi qthreads@1.16     5axnc7s swig@4.0.2
sbw7ukx gasnet@2021.3.0   x3kwocp libsigsegv@2.12    mbedwdv py-numpy@1.20.3           vmnfg3u py-setuptools@50.3.2   pdl6hkb raja@0.13.0       pmkt2f7 swig@4.0.2-fortran
p4itgk3 ginkgo@1.3.0      meebxkm libxml2@2.9.10     fiou22u py-periodictable@1.5.0    hpeak4v py-toml@0.10.2         lueanor sqlite@3.34.0     o4jw6v3 umpire@4.1.2
36pueen hdf5@1.8.22       oguh3so m4@1.4.18          iqa4mv6 py-petsc4py@3.15.0        yohkwul py-warpx@21.05         wmwpoyc strumpack@5.1.1   ufqtz47 upcxx@2021.3.0
v5mkktz hdf5@1.10.7       2iqgyya mfem@4.2.0         vhk4hzt py-picmistandard@develop  2dopy2n py-warpx@21.05         utolp4n sundials@5.7.0
```

Running Tests via spack test run

```
siddiq90@cori04> spack test run hdf5
==> Spack test st5uy2i326u2y7syw4wwmrpwk2233wta
==> Testing package hdf5-1.8.22-36pueen
==> Testing package hdf5-1.10.7-v5mkktz

siddiq90@cori04> spack test results st5uy2i326u2y7syw4wwmrpwk2233wta
==> Results for test suite 'st5uy2i326u2y7syw4wwmrpwk2233wta':
==>   hdf5-1.8.22-36pueen PASSED
==>   hdf5-1.10.7-v5mkktz PASSED
```

Test Suite name

```
siddiq90@cori04> spack test run --alias py-warpx py-warpx
==> Spack test py-warpx
==> Testing package py-warpx-21.05-yohkwul
==> Testing package py-warpx-21.05-2dopy2n
==> Testing package py-warpx-21.05-ownfza6

siddiq90@cori04> spack test results py-warpx
==> Results for test suite 'py-warpx':
==>   py-warpx-21.05-yohkwul PASSED
==>   py-warpx-21.05-2dopy2n PASSED
==>   py-warpx-21.05-ownfza6 PASSED
```

Retrieve Test Results via spack test results

```
siddiq90@cori04> spack test results
==> Results for test suite 'st5uy2i326u2y7syw4wwmrpwk2233wta':
==>   hdf5-1.8.22-36pueen PASSED
==>   hdf5-1.10.7-v5mkktz PASSED
==> Results for test suite 'zevmet73tlterufszagrkesihnjox7kr':
==>   m4-1.4.18-oguh3so PASSED
```

```
siddiq90@cori04> spack test results -- m4
==> Results for test suite 'zevmet73tlterufszagrkesihnjox7kr', spec matching 'm4':
==>   m4-1.4.18-oguh3so PASSED
```

```
siddiq90@cori04> spack test results -l -- m4
==> Results for test suite 'zevmet73tlterufszagrkesihnjox7kr', spec matching 'm4':
==>   m4-1.4.18-oguh3so PASSED
==> Testing package m4-1.4.18-oguh3so
==> [2021-08-13-10:15:32.751155] test: ensuring m4 version is 1.4.18
==> [2021-08-13-10:15:32.753276] '/global/common/software/spackecp/e4s-21.05/software/cray-cnl7-haswell/intel-19.1.3.304/m4-1.4.18-oguh3soitsu3ym3fb4pdy7p7jfthl7am/bin/m4'
 '--version'
m4 (GNU M4) 1.4.18
Copyright (C) 2016 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>.
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.

Written by Rene' Seindal.
PASSED
==> [2021-08-13-10:15:32.847466] test: ensuring m4 example succeeds
==> [2021-08-13-10:15:32.848512] '/global/common/software/spackecp/e4s-21.05/software/cray-cnl7-haswell/intel-19.1.3.304/m4-1.4.18-oguh3soitsu3ym3fb4pdy7p7jfthl7am/bin/m4'
 '/global/homes/s/siddiq90/.spack/test/zevmet73tlterufszagrkesihnjox7kr/m4-1.4.18-oguh3so/data/m4/hello.m4'

// macro is defined
Hello, World!
PASSED
```

Facility Testing Use Case

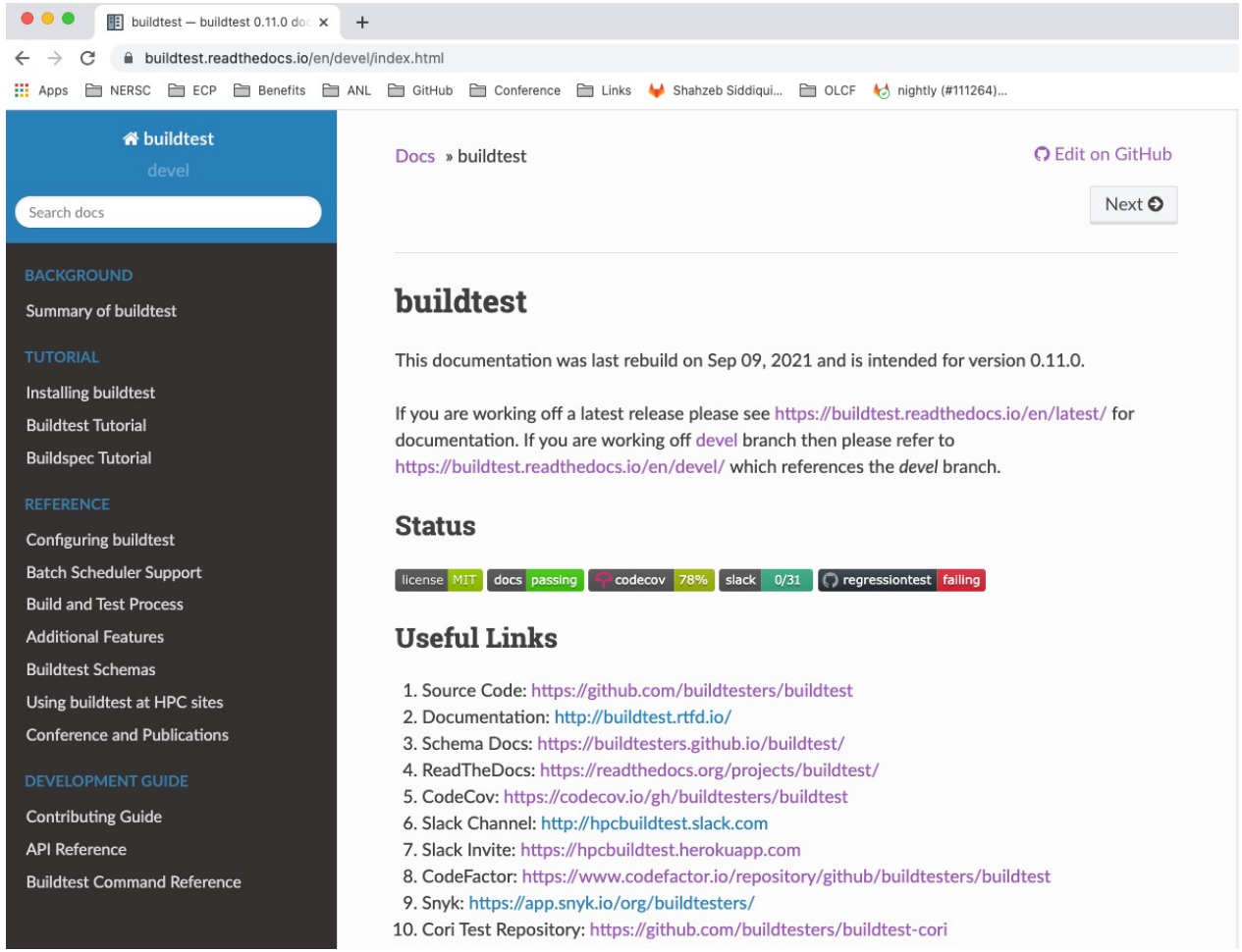
- We need comprehensive system and software level testing
- We need to test facility deployment of e4s stacks that are usually tied to fix version of spack.
- Test system layer – configuration, filesystem, job scheduler, drivers, modules
- Run benchmark
- Negative tests to address known bugs in system
- User provided tests as part of User Support Tickets

What is buildtest

- Buildtest is a testing framework that builds and execute tests on your HPC systems
- Buildtest is intended for HPC staff, developers to build a **Facility Testsuite** for their HPC systems
- Tests are written in YAML called **buildspecs** which buildtest process to generate shell scripts.
- Buildspecs are validated with JSON schema.
- Support test execution on local machine or via batch schedulers. Currently we support Slurm, LSF, PBS and Cobalt.
- The framework is implemented in python
- Available on GitHub at https://github.com/buildtesters/buildtest

Installation

```
git clone https://github.com/buildtesters/buildtest.git
cd buildtest
source setup.sh
```



Project Summary

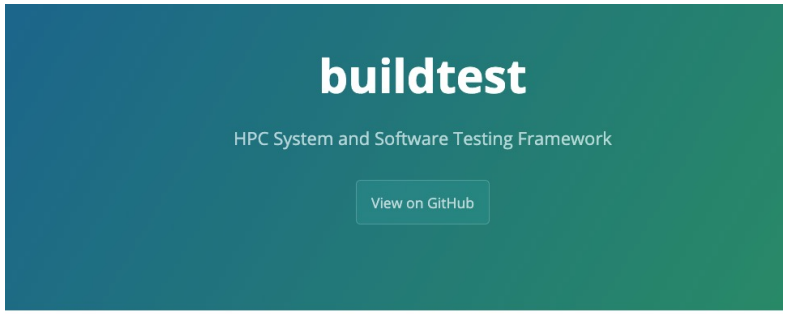
- Total of 37 releases
- Added spack support in v0.10.0
- Added support for Slurm and LSF in v0.8.0, Cobalt in v0.9.1 and PBS in v0.9.5
- Initially a bash program which was converted to python 2 and eventually migrated to python 3.
- Release updates are documented in CHANGELOG.rst
- Distributed as MIT License
- Documentation is built using sphinx and hosted via readthedocs platform

Design Goals

- Perform component level testing for system and software stack
- Provide a standard template for writing tests
- Abstract low-level system configuration
- Framework should automate build and execution of test
- Framework must support local and batch submission test

Schemas

- The schema development is implemented independent to buildtest. The schemas and docs are hosted at https://buildtesters.github.io/buildtest/
- We run regression test against example YAML files for each schema to ensure schemas are written in accordance to desired YAML construct.
- We automate JSON Schema documentation using adobe/jsonschema2md into Markdown pages and publish schema and documentation to GitHub pages
- Schemas are versioned to allow development to schemas and its YAML structure.



Buildtest Schema

This repository contains the schemas used by buildtest.

buildtest schema docs can be found at https://buildtesters.github.io/buildtest/

Currently, we support the following schemas:

- definitions: This schema definitions JSON definitions that are referenced by other schemas.
- global: The global schema inherited by all sub-schemas
- compiler-v1.0: Compiler sub-schema version 1.0 using `type: compiler`
- script-v1.0: Script sub-schema version 1.0 using `type: script`
- settings: This schema defines the content of buildtest settings file to configure buildtest.

The schemas are published at https://github.com/buildtesters/buildtest/tree/gh-pages/pages/devel/schemas

compiler schema version 1.0 Schema

`compiler-v1.0.schema.json`

The compiler schema is of `type: compiler` in sub-schema which is used for compiling and running programs

Abstract	Extensible	Status	Identifiable	Custom Properties	Additional Properties	Acc Restri
Can be instantiated	Yes	Unknown status	No	Forbidden	Forbidden	none

compiler schema version 1.0 Type

`object` (compiler schema version 1.0)

compiler schema version 1.0 Properties

Property	Type	Required	Nullable	Defined by
type	`string`	Required	cannot be null	compiler schema version 1.0
description	`string`	Optional	cannot be null	compiler schema version 1.0
compilers	`object`	Required	cannot be null	compiler schema version 1.0
source	`string`	Required	cannot be null	compiler schema version 1.0
executor	`string`	Required	cannot be null	compiler schema version 1.0
run_only	`object`	Optional	cannot be null	compiler schema version 1.0
skip	`boolean`	Optional	cannot be null	compiler schema version 1.0
tags	Merged	Optional	cannot be null	compiler schema version 1.0

type

Select schema type to use when validating buildspec. This must be of set to `compiler`.

`type`

- is required
- Type: `string`
- cannot be null
- defined in: compiler schema version 1.0

type Type

`string`

type Constraints

pattern: the string must match the following regular expression:

`^compiler$`

try pattern

Preview of buildtest

```
(buildtest) bash-3.2$ buildtest help build

Building Buildspecs
--------------------

Command                                                 Description

buildtest build -b <file>                               Build a single buildspec file
buildtest build -b <dir>                                Build all buildspecs recursively in a given directory
buildtest build -b <file> -b <dir>                      Build buildspecs by file and directory
buildtest build -b <file> -b <dir> -x <file> -x <dir>   Exclude files and directory when building buildspecs
buildtest build -t pass  -t python                      Build buildspecs by tagname 'pass' and 'python'
buildtest build -e <executor1> -e <executor2>           Building buildspecs by executor
buildtest build -b <file> -t <tagname1> -e <executor1>  Building buildspecs with file, directory, tags, and executors
buildtest build -b tutorials  --filter type=script      Build all tests in directory 'tutorials' and filter tests by type='script'
buildtest build -b tutorials  --filter tags=pass        Build all tests in directory 'tutorials' and filter tests by tags='pass'
buildtest build -b tutorials  --filter maintainers=@bob Build all tests in directory 'tutorials' and filter tests by maintainers='@bob'
buildtest build --helpfilter                            Show list of filter fields used with --filter option
buildtest -c config.yml build -b <file>                 Use buildtest configuration file 'config.yml'
buildtest build -b <file> --rebuild 5                   Rebuild a test 5 times
buildtest build -b <file> --testdir /tmp                Write tests in /tmp
```

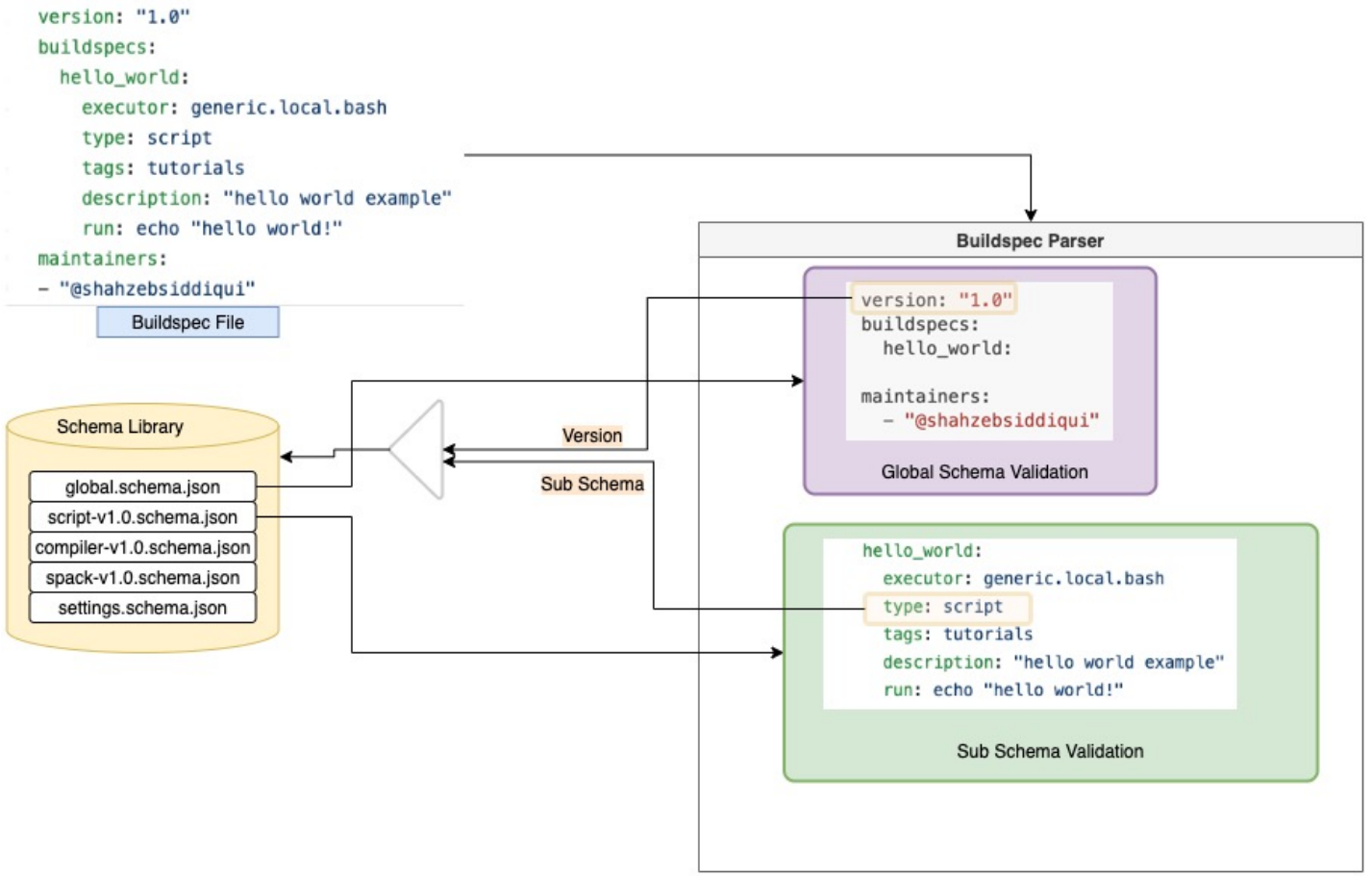
General Pipeline

- Discover: Find buildspecs based on search criteria (file, directory, tags, executor)
- Parse: Validates buildspec with JSON Schema
- Build: Generates testscript from YAML
- Run: Executes tests via local or batch executor and retrieve return code and output/error file.
- Update Report: Update report file with test results including any metadata

Parse: Buildspec Validation Process

- Every buildspec is validated by global schema and a subschema defined by **type** field.
- Buildtest will skip any buildspecs that fails validation.

```
"$id": "global.schema.json",
"$schema": "http://json-schema.org/draft-07/schema#",
"title": "global schema",
"description": "buildtest global schema is validated for all buildspecs.
"type": "object",
"required": ["version","buildspecs"],
```



Demo – Buildtest Tutorial

Buildspec Script Schema

```
version: "1.0"                                                    Schema Version
buildspecs:                                                       Declaration of tests
  systemd_default_target:                                         Name of Test
    executor: generic.local.bash                                  Name of Executor
    type: script                                                  Schema Type
    tags: [system]                                                Tag Name
    description: check if default target is multi-user.target     Description of Test
    run: |
      if [ "multi-user.target" == `systemctl get-default` ]; then
        echo "multi-user is the default target";
        exit 0
      fi
      echo "multi-user is not the default target";
      exit 1
```

Script

```
"$id": "script-v1.0.schema.json",
"$schema": "http://json-schema.org/draft-07/schema#",
"title": "script schema version 1.0",
"description": "The script schema is of ``type: script`` in sub-schema which is used for running shell scripts",
"type": "object",
"required": ["type", "run", "executor"],
"additionalProperties": false,
```

Status Check – Regular Expression

- Buildtest supports status check of test based on regular expression, returncode and runtime. This can be configured via **status** property

```
+----------------------+
| Stage: Test Summary  |
+----------------------+

 name              | id       | executor           | status   | returncode_match   | regex_match   | runtime_match   |   returncode |   runtime
-------------------+----------+--------------------+----------+--------------------+---------------+-----------------+--------------+-----------
 status_regex_pass | f4bbe4e6 | generic.local.bash | PASS     | False              | True          | False           |            0 |  0.802642
 status_regex_fail | 3fcfdf3d | generic.local.bash | FAIL     | False              | False         | False           |            0 |  0.990494


Passed Tests: 1/2 Percentage: 50.000%
Failed Tests: 1/2 Percentage: 50.000%
```

```yaml
version: "1.0"
buildspecs:
  status_regex_pass:
    executor: generic.local.bash
    type: script
    tags: [system]
    description: Pass test based on regular expression
    run: echo "PASS"
    status:
      regex:
        stream: stdout
        exp: "^(PASS)$"

  status_regex_fail:
    executor: generic.local.bash
    type: script
    tags: [system]
    description: Pass test based on regular expression
    run: echo "FAIL"
    status:
      regex:
        stream: stdout
        exp: "^(123FAIL)$"
```

Status Check – Return Code

- The return code field can be used to customize how test is passed, by default a return code 0 is a **PASS.** The return code can be a single number or a list of return codes to match.

```
version: "1.0"
buildspecs:

  exit1_fail:
    executor: generic.local.sh
    type: script
    description: exit 1 by default is FAIL
    tags: [tutorials, fail]
    run: exit 1

  exit1_pass:
    executor: generic.local.sh
    type: script
    description: report exit 1 as PASS
    run: exit 1
    tags: [tutorials, pass]
    status:
      returncode: [1]

  returncode_list_mismatch:
    executor: generic.local.sh
    type: script
    description: exit 2 failed since it failed to match returncode 1
    run: exit 2
    tags: [tutorials, fail]
    status:
      returncode: [1, 3]

  returncode_int_match:
    executor: generic.local.sh
    type: script
    description: exit 128 matches returncode 128
    run: exit 128
    tags: [tutorials, pass]
    status:
      returncode: 128
```

```
+----------------------+
| Stage: Test Summary  |
+----------------------+

 name                    | id       | executor          | status | returncode_match | regex_match | runtime_match |  returncode |   runtime
-------------------------+----------+-------------------+--------+------------------+-------------+---------------+-------------+-----------
 exit1_fail              | c75bbee6 | generic.local.sh  | FAIL   | N/A              | N/A         | N/A           |           1 |  0.840106
 exit1_pass              | a1ca70c3 | generic.local.sh  | PASS   | True             | False       | False         |           1 |  0.643247
 returncode_list_mismatch | 19eed66c | generic.local.sh  | FAIL   | False            | False       | False         |           2 |  0.429428
 returncode_int_match    | a1ec126f | generic.local.sh  | PASS   | True             | False       | False         |         128 |  0.486569


Passed Tests: 2/4 Percentage: 50.000%
Failed Tests: 2/4 Percentage: 50.000%
```

Status Check - Runtime

- Buildtest can determine PASS/FAIL based on test runtime. This can be specified using **runtime** property with options for specifying **min** or **max** or both if one wants to set a range.

```
+----------------------+
| Stage: Test Summary  |
+----------------------+
```

name	id	executor	status	returncode_match	regex_match	runtime_match	returncode	runtime
timelimit_min	dca06490	generic.local.sh	PASS	False	False	True	0	2.67952
timelimit_min_max	0a6b1d07	generic.local.sh	PASS	False	False	True	0	2.83349
timelimit_max	2ae3c8f6	generic.local.sh	PASS	False	False	True	0	2.28634
timelimit_min_fail	83eea676	generic.local.sh	FAIL	False	False	False	0	2.33904
timelimit_max_fail	d8aad7ca	generic.local.sh	FAIL	False	False	False	0	3.25817

```yaml
version: "1.0"
buildspecs:
  timelimit_min_max:
    type: script
    executor: generic.local.sh
    description: "Run a sleep job for 2 seconds and test pass if its within 1.0-3.0sec"
    tags: ["tutorials"]
    run: sleep 2
    status:
      runtime:
        min: 1.0
        max: 3.0


  timelimit_min:
    type: script
    executor: generic.local.sh
    description: "Run a sleep job for 2 seconds and test pass if its exceeds min time of 1.0 se
    tags: ["tutorials"]
    run: sleep 2
    status:
      runtime:
        min: 1.0

  timelimit_max:
    type: script
    executor: generic.local.sh
    description: "Run a sleep job for 2 seconds and test pass if it's within max time: 5.0 sec"
    tags: ["tutorials"]
    run: sleep 2
    status:
      runtime:
        max: 5.0


  timelimit_min_fail:
    type: script
    executor: generic.local.sh
    description: "This test fails because it runs less than mintime of 10 second"
    tags: ["tutorials"]
    run: sleep 2
    status:
      runtime:
        min: 10.0

  timelimit_max_fail:
    type: script
    executor: generic.local.sh
    description: "This test fails because it exceeds maxtime of 1.0 second"
    tags: ["tutorials"]
    run: sleep 3
    status:
      runtime:
        max: 1.0
```

Multi Executors

- Every test must be assigned to an executor that is responsible for running test. This is specified via **executor** property however one can specify a regular expression to run across multiple executors.
- The **executors** property can be used to define configuration based on executor that are specific to each test run.
- The **vars** and **env** are used for declaring variables and environment variables which expects a list of Key/Value pair
- Executors are defined in your configuration file which can be retrieved via **buildtest config executors**
- In this example we run a single test with executor **generic.local.sh** and **generic.local.bash**

```
(buildtest) bash-3.2$ buildtest config executors
generic.local.bash
generic.local.sh
generic.local.csh
generic.local.zsh
generic.local.python
```

```yaml
version: "1.0"
buildspecs:
  executors_vars_env_declaration:
    type: script
    executor: 'generic.local.(bash|sh)'
    description: Declaring env and vars by executors section
    tags: [tutorials]
    run: |
      echo "X:" $X
      echo "Y:" $Y
      echo $SHELL

    executors:
      generic.local.bash:
        vars:
          X: 1
          Y: 3
        env:
          SHELL: bash
      generic.local.sh:
        vars:
          X: 2
          Y: 4
        env:
          SHELL: sh
```

```
+----------------------+
| Stage: Test Summary  |
+----------------------+
```

name	id	executor	status	returncode_match	regex_match	runtime_match	returncode	runtime
executors_vars_env_declaration	ba119b8b	generic.local.bash	PASS	N/A	N/A	N/A	0	0.618133
executors_vars_env_declaration	b06ca510	generic.local.sh	PASS	N/A	N/A	N/A	0	0.808272

Buildspec Compiler Schema

```
"$id": "compiler-v1.0.schema.json",
"$schema": "http://json-schema.org/draft-07/schema#",
"title": "compiler schema version 1.0",
"description": "The compiler schema is of ``type: compiler``
"type": "object",
"required": [
  "type",
  "source",
  "compilers",
  "executor"
],
```

- The compiler schema is used for compiling single source code with compilers
- You must use **type: compiler** to define tests using this schema
- This test will be built with all gcc compilers
- Compilers are defined in buildtest configuration, one can retrieve compilers using **buildtest config compilers**

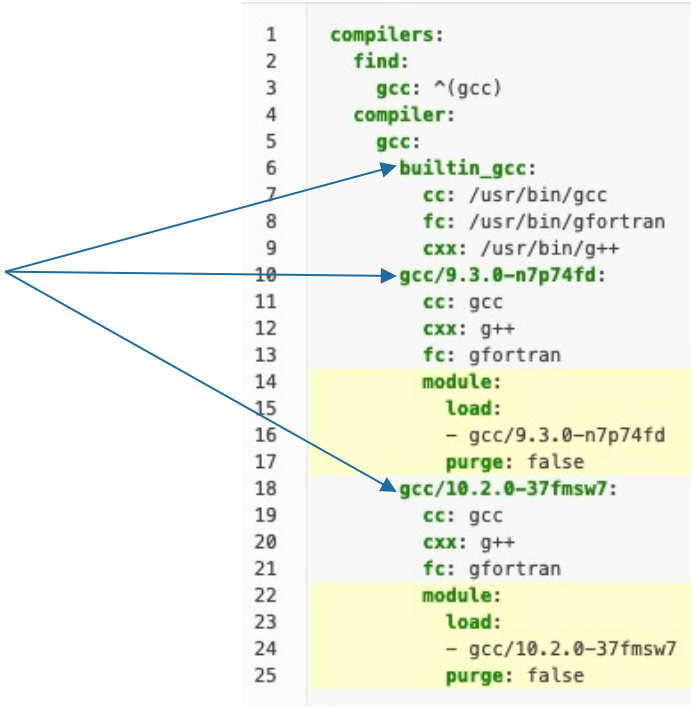
```
$ buildtest config compilers -l
builtin_gcc
gcc/9.3.0-n7p74fd
gcc/10.2.0-37fmsw7
```

```
version: "1.0"
buildspecs:
  vecadd_gnu:
    type: compiler                                                    Compiler Schema
    description: Vector Addition example with GNU compiler
    tags: [tutorials, compile]
    executor: generic.local.bash
    source: src/vecAdd.c                                              Source File
    compilers:                                                        Start of Compiler Block
      name: ["^(builtin_gcc|gcc)"]                                    Select Compilers based on Regular Expression
      default:                                                        Default Section for compilers organized by compiler groups
        gcc:                                                          Default Section for gcc compilers
          cflags: -fopenacc                                           Set cflags
          ldflags: -lm                                                Set ldflags
```

Override Compiler Default

- Compiler defaults can be overridden in **config** section which expects compiler names defined in buildtest setting.
- Buildtest will ignore compiler in **config** if it's not picked up in regular expression.

Compiler Names



```yaml
compilers:
  find:
    gcc: ^(gcc)
  compiler:
    gcc:
      builtin_gcc:
        cc: /usr/bin/gcc
        fc: /usr/bin/gfortran
        cxx: /usr/bin/g++
      gcc/9.3.0-n7p74fd:
        cc: gcc
        cxx: g++
        fc: gfortran
        module:
          load:
          - gcc/9.3.0-n7p74fd
          purge: false
      gcc/10.2.0-37fmsw7:
        cc: gcc
        cxx: g++
        fc: gfortran
        module:
          load:
          - gcc/10.2.0-37fmsw7
          purge: false
```

```yaml
version: "1.0"
buildspecs:
  hello_c:
    type: compiler
    description: "Hello World C Compilation"
    executor: generic.local.bash
    tags: [tutorials, compile]
    source: "src/hello.c"
    compilers:
      name: ["^(builtin_gcc|gcc)"]
      default:
        gcc:
          cflags: -O1
      config:
        gcc/9.3.0-n7p74fd:
          cflags: -O2
        gcc/10.2.0-37fmsw7:
          cflags: -O3
```

Scheduler Support

Slurm

```yaml
version: "1.0"
buildspecs:
  slurm_metadata:
    description: Get metadata from compute node when submitting job
    type: script
    executor: cori.slurm.debug
    sbatch:
      - "-t 00:05"
      - "-C haswell"
      - "-N 1"
    run: |
      export SLURM_JOB_NAME="firstjob"
      echo "jobname:" $SLURM_JOB_NAME
      echo "slurmdb host:" $SLURMD_NODENAME
      echo "pid:" $SLURM_TASK_PID
      echo "submit host:" $SLURM_SUBMIT_HOST
      echo "nodeid:" $SLURM_NODEID
      echo "partition:" $SLURM_JOB_PARTITION
```

LSF

```yaml
version: "1.0"
buildspecs:
  hostname:
    type: script
    executor: ascent.lsf.batch
    bsub: [ "-W 10",  "-nnodes 1"]

    run: jsrun hostname
```

Cobalt

```yaml
version: "1.0"
buildspecs:
  yarrow_hostname:
    executor: jlse.cobalt.yarrow
    type: script
    cobalt: ["-n 1", "--proccount 1", "-t 10"]
    run: hostname
```

PBS

```yaml
version: "1.0"
buildspecs:
  pbs_sleep:
    type: script
    executor: generic.pbs.workq
    pbs: ["-l nodes=1", "-l walltime=00:02:00"]
    run: sleep 10
```

```
+---------------------+
| Stage: Running Test |
+---------------------+


_______________________________
Launching test: pbs_sleep
Test ID: 5c8cc6cd-51d6-451e-88bd-aa251e177781
Executor Name: generic.pbs.workq
Running Test:  /tmp/GitHubDesktop/buildtest/var/tests/generic.pbs.workq/sleep/pbs_sleep/5c8cc6cd/pbs_sleep_build.sh
[pbs_sleep] JobID: 396.pbs dispatched to scheduler
Polling Jobs in 5 seconds


Current Jobs
---------------


+-----------+----------+-------------------+---------+----------+---------+
|   name    |    id    |     executor      |  jobID  | jobstate | runtime |
+-----------+----------+-------------------+---------+----------+---------+
| pbs_sleep | 5c8cc6cd | generic.pbs.workq | 396.pbs |    R     |  5.245  |
+-----------+----------+-------------------+---------+----------+---------+
Polling Jobs in 5 seconds
pbs_sleep/5c8cc6cd: Job 396.pbs is complete!
pbs_sleep/5c8cc6cd: Writing output file: /tmp/GitHubDesktop/buildtest/var/tests/generic.pbs.workq/sleep/pbs_sleep/5c8cc6cd/pbs_sleep.o396
pbs_sleep/5c8cc6cd: Writing error file: /tmp/GitHubDesktop/buildtest/var/tests/generic.pbs.workq/sleep/pbs_sleep/5c8cc6cd/pbs_sleep.e396

+----------------------+
| Completed Polled Jobs |
+----------------------+

 name      | id       | executor          | jobID   | jobstate   | status   |   returncode |   runtime
-----------+----------+-------------------+---------+------------+----------+--------------+-----------
 pbs_sleep | 5c8cc6cd | generic.pbs.workq | 396.pbs | F          | PASS     |            0 |   10.2916

+----------------------+
| Stage: Test Summary  |
+----------------------+

 name      | id       | executor          | status   | returncode_match   | regex_match   | runtime_match   |   returncode |   runtime
-----------+----------+-------------------+----------+--------------------+---------------+-----------------+--------------+-----------
 pbs_sleep | 5c8cc6cd | generic.pbs.workq | PASS     | N/A                | N/A           | N/A             |            0 |   10.2916


Passed Tests: 1/1 Percentage: 100.000%
Failed Tests: 0/1 Percentage: 0.000%
```

EXASCALE COMPUTING PROJECT

Max Pend Time and Poll Interval

- Buildtest will poll batch jobs at set interval to get updated job state for all jobs in queue, once job is complete buildtest will gather job results and metadata of job.
- The **pollinterval** property configures number of seconds to sleep until we poll jobs for updated job state. This value can be overridden on command line via buildtest build –poll-interval
- Buildtest will cancel pending or suspended jobs after pending time exceeds **max_pend_time**. This value can be overridden via buildtest build –max-pend-time

```
+--------------------+
| Stage: Running Test |
+--------------------+


_______________________________
Launching test: pbs_hold_job
Test ID: b5138507-6f36-4451-b58c-2f600c528dbb
Executor Name: generic.pbs.workq
Running Test:  /tmp/GitHubDesktop/buildtest/var/tests/generic.pbs.workq/hold/pbs_hold_job/b5138507/pbs_hold_job_build.sh
[pbs_hold_job] JobID: 391.pbs dispatched to scheduler
Polling Jobs in 8 seconds


Current Jobs
---------------


+--------------+----------+--------------------+---------+----------+---------+
|     name     |    id    |     executor       |  jobID  | jobstate | runtime |
+--------------+----------+--------------------+---------+----------+---------+
| pbs_hold_job | b5138507 | generic.pbs.workq | 391.pbs |    H     |  8.158  |
+--------------+----------+--------------------+---------+----------+---------+
Polling Jobs in 8 seconds
pbs_hold_job/b5138507: Cancelling Job: 391.pbs because job exceeds max pend time: 10 sec with current pend time of 16.208

Cancelled Jobs: [pbs_hold_job/b5138507]
Unable to run any tests
```

```yaml
executors:
  defaults:
    pollinterval: 10
    launcher: sbatch
    max_pend_time: 90
    account: nstaff
```

Demo – Buildspecs Tutorial

Filter and Format buildspec cache

- We can filter and format buildspec cache using –filter and –format option.
- The filter option expects a list of key=value pair separated by comma.
- To see list of all filter and format fields we can use –helpfilter and –helpformat option

```
$ buildtest buildspec find --helpfilter

Field     Description                Type
--------  -------------------------  ------
executor  Filter by executor name    STRING
tags      Filter by tag name         STRING
type      Filter by schema type      STRING
```

```
$ buildtest buildspec find --helpformat
Field        Description
-----------  ----------------------
description  Format by description
executor     Format by executor type
file         Format by file
name         Format by test name
tags         Format by tag name
type         Format by schema type
```

```
$ buildtest buildspec find --filter tags=fail

+--------------------------+--------+-----------------+----------------------+-----------------------------------------------------+
| name                     | type   | executor        | tags                 | description                                         |
+==========================+========+=================+======================+=====================================================+
| exit1_fail               | script | generic.local.sh | ['tutorials', 'fail'] | exit 1 by default is FAIL                           |
+--------------------------+--------+-----------------+----------------------+-----------------------------------------------------+
| returncode_list_mismatch | script | generic.local.sh | ['tutorials', 'fail'] | exit 2 failed since it failed to match returncode 1 |
+--------------------------+--------+-----------------+----------------------+-----------------------------------------------------+
```

```
$ buildtest buildspec find --filter tags=fail --format name,tags

+--------------------------+----------------------+
| name                     | tags                 |
+==========================+======================+
| exit1_fail               | ['tutorials', 'fail'] |
+--------------------------+----------------------+
| returncode_list_mismatch | ['tutorials', 'fail'] |
+--------------------------+----------------------+
```

Multi key filter is evaluated as logical AND.

```
$ buildtest buildspec find --filter tags=tutorials,executor=generic.local.sh,type=script

+--------------------------+--------+-----------------+----------------------+-----------------------------------------------------+
| name                     | type   | executor        | tags                 | description                                         |
+==========================+========+=================+======================+=====================================================+
| _bin_sh_shell            | script | generic.local.sh | ['tutorials']        | /bin/sh shell example                               |
+--------------------------+--------+-----------------+----------------------+-----------------------------------------------------+
| sh_shell                 | script | generic.local.sh | ['tutorials']        | sh shell example                                    |
+--------------------------+--------+-----------------+----------------------+-----------------------------------------------------+
| shell_options            | script | generic.local.sh | ['tutorials']        | shell options                                       |
+--------------------------+--------+-----------------+----------------------+-----------------------------------------------------+
| exit1_fail               | script | generic.local.sh | ['tutorials', 'fail'] | exit 1 by default is FAIL                           |
+--------------------------+--------+-----------------+----------------------+-----------------------------------------------------+
| exit1_pass               | script | generic.local.sh | ['tutorials', 'pass'] | report exit 1 as PASS                               |
+--------------------------+--------+-----------------+----------------------+-----------------------------------------------------+
| returncode_list_mismatch | script | generic.local.sh | ['tutorials', 'fail'] | exit 2 failed since it failed to match returncode 1 |
+--------------------------+--------+-----------------+----------------------+-----------------------------------------------------+
| returncode_int_match     | script | generic.local.sh | ['tutorials', 'pass'] | exit 128 matches returncode 128                     |
+--------------------------+--------+-----------------+----------------------+-----------------------------------------------------+
```

Show content of buildspec

- The **buildtest buildspec show** command can show content of buildspec based on a given test name. In this example we show content of test python_hello

```
(buildtest) bash-3.2$ buildtest buildspec show python_hello
version: "1.0"
buildspecs:
  python_hello:
    type: script
    description: Hello World python
    executor: generic.local.bash
    tags: python
    run: python hello.py


buildspec:  /Users/siddiq90/Documents/GitHubDesktop/buildtest/tutorials/python-hello.yml
```

Validate Buildspecs

- The buildtest buildspec validate can be used to validate buildspecs with JSON schema and command options mimic similar to buildtest build

```
(buildtest) bash-3.2$ buildtest buildspec validate -t python
Processing buildspec: /Users/siddiq90/Documents/GitHubDesktop/buildtest/tutorials/python-shell.yml
Processing buildspec: /Users/siddiq90/Documents/GitHubDesktop/buildtest/tutorials/python-hello.yml
All buildspecs passed validation!!!
```

```
(buildtest) bash-3.2$ buildtest buildspec validate -b tutorials/invalid_tags.yml

file:  /Users/siddiq90/Documents/GitHubDesktop/buildtest/tutorials/invalid_tags.yml
--------------------------------------------------------------------------------
['network', 'network'] is not valid under any of the given schemas

Failed validating 'oneOf' in schema['properties']['tags']:
    {'oneOf': [{'type': 'string'},
               {'$ref': '#/definitions/list_of_strings'}]}

On instance['tags']:
    ['network', 'network']


Processing buildspec: /Users/siddiq90/Documents/GitHubDesktop/buildtest/tutorials/invalid_tags.yml
There were 1 buildspecs that failed validation
```

Show all invalid buildspecs

- Buildtest will keep record of all invalid buildspecs in the cache upon running buildtest buildspec find, you can retrieve a list of all invalid buildspecs via buildtest buildspec find invalid command.
- The **-e** option will print all error messages for every invalid buildspecs
- If you want to load all buildspecs in cache and fix invalid buildspecs then **buildtest buildspec find invalid** would be appropriate, however if you want to validate any buildspec without loading in cache you can use **buildtest buildspec validate**

```
(buildtest) bash-3.2$ buildtest buildspec find invalid
+-----------------------------------------------------------------------------------------------------------+
| buildspecs                                                                                                |
+===========================================================================================================+
| /Users/siddiq90/Documents/GitHubDesktop/buildtest/tutorials/invalid_buildspec_section.yml                 |
+-----------------------------------------------------------------------------------------------------------+
| /Users/siddiq90/Documents/GitHubDesktop/buildtest/tutorials/invalid_tags.yml                              |
+-----------------------------------------------------------------------------------------------------------+
| /Users/siddiq90/Documents/GitHubDesktop/buildtest/tutorials/invalid_executor.yml                          |
+-----------------------------------------------------------------------------------------------------------+
| /Users/siddiq90/Documents/GitHubDesktop/buildtest/tutorials/burstbuffer_datawarp_executors.yml            |
+-----------------------------------------------------------------------------------------------------------+
| /Users/siddiq90/Documents/GitHubDesktop/buildtest/tutorials/spack/env_install.yml                         |
+-----------------------------------------------------------------------------------------------------------+
| /Users/siddiq90/Documents/GitHubDesktop/buildtest/tutorials/spack/spack_multiple_executor_sbatch.yml      |
+-----------------------------------------------------------------------------------------------------------+
| /Users/siddiq90/Documents/GitHubDesktop/buildtest/general_tests/sched/pbs/batch.yml                       |
+-----------------------------------------------------------------------------------------------------------+
| /Users/siddiq90/Documents/GitHubDesktop/buildtest/general_tests/sched/pbs/hostname.yml                    |
+-----------------------------------------------------------------------------------------------------------+
```

```
(buildtest) bash-3.2$ buildtest buildspec find invalid -e
/Users/siddiq90/Documents/GitHubDesktop/buildtest/tutorials/invalid_buildspec_section.yml
_____________________________________________________________________________
'[/Users/siddiq90/Documents/GitHubDesktop/buildtest/tutorials/invalid_buildspec_section.yml]: type badscript is not known to buildtest.'
_____________________________________________________________________________


/Users/siddiq90/Documents/GitHubDesktop/buildtest/tutorials/invalid_tags.yml
_____________________________________________________________________________
['network', 'network'] is not valid under any of the given schemas

Failed validating 'oneOf' in schema['properties']['tags']:
    {'oneOf': [{'type': 'string'},
               {'$ref': '#/definitions/list_of_strings'}]}

On instance['tags']:
    ['network', 'network']
_____________________________________________________________________________
```

Invalid value for **type** field

Can't have duplicate tag names for tags property

Report Summary

```
(buildtest) bash-3.2$ buildtest report summary
Report:  /Users/siddiq90/Documents/GitHubDesktop/buildtest/var/report.json
Total Tests: 6
Total Tests by Names:  6
Number of buildspecs in report:  3
```

Breakdown by Test

name	runs	pass	fail
variables_bash	1	1	0
sleep	1	1	0
exit1_fail	1	0	1
exit1_pass	1	1	0
returncode_list_mismatch	1	0	1
returncode_int_match	1	1	0

FAIL test

name	id	executor	state	returncode	runtime
exit1_fail	996d9773	generic.local.sh	FAIL	1	0.098565
returncode_list_mismatch	6049eea8	generic.local.sh	FAIL	2	0.107125

Query Test Reports with Filter and Format Examples

- We provide access to test reports through CLI. The reports are stored in JSON file for post-processing.
- The buildtest report will display all test results which can be queried with filter and format options.
- The **–filter** option are passed as key=value pair
- Multiple filter arguments can be delimited by comma separator and buildtest will treat multiple filter argument as a logical AND operation
- The **–format** option alter the columns in the report tables.

```
$ buildtest report --filter state=PASS,executor=generic.local.python --format=name,id,state,executor

+-------------------------+----------+---------+---------------------+
| name                    | id       | state   | executor            |
+=========================+==========+=========+=====================+
| circle_area             | 4d875c18 | PASS    | generic.local.python |
+-------------------------+----------+---------+---------------------+
| circle_area             | 74864a25 | PASS    | generic.local.python |
+-------------------------+----------+---------+---------------------+
| circle_area             | 882ea4ac | PASS    | generic.local.python |
+-------------------------+----------+---------+---------------------+
| circle_area             | 447e5125 | PASS    | generic.local.python |
+-------------------------+----------+---------+---------------------+
| circle_area             | 015ec352 | PASS    | generic.local.python |
+-------------------------+----------+---------+---------------------+
| run_only_platform_darwin | 62ed7919 | PASS    | generic.local.python |
+-------------------------+----------+---------+---------------------+
| run_only_platform_darwin | bdaec723 | PASS    | generic.local.python |
+-------------------------+----------+---------+---------------------+
```

```
$ buildtest report --filter returncode=1 --format=name,id,returncode

+-------------------------+----------+--------------+
| name                    | id       |   returncode |
+=========================+==========+==============+
| systemd_default_target  | fbfaa7d4 |            1 |
+-------------------------+----------+--------------+
| systemd_default_target  | dd0e2462 |            1 |
+-------------------------+----------+--------------+
| systemd_default_target  | c72a1f98 |            1 |
+-------------------------+----------+--------------+
| systemd_default_target  | 5920ee28 |            1 |
+-------------------------+----------+--------------+
| exit1_fail              | b8c5fde0 |            1 |
+-------------------------+----------+--------------+
| exit1_fail              | f1a3ad7d |            1 |
+-------------------------+----------+--------------+
```

```
$ buildtest report --filter name=exit1_pass --format=name,id,returncode,state

+------------+----------+--------------+---------+
| name       | id       |   returncode | state   |
+============+==========+==============+=========+
| exit1_pass | e6933549 |            1 | PASS    |
+------------+----------+--------------+---------+
| exit1_pass | 60d26637 |            1 | PASS    |
+------------+----------+--------------+---------+
| exit1_pass | 50e730ee |            1 | PASS    |
+------------+----------+--------------+---------+
| exit1_pass | 105b838e |            1 | PASS    |
+------------+----------+--------------+---------+
```

Format And Filter fields for buildtest report

- The **buildtest report** command provides a description of format and filter fields using **–helpformat** and **–helpfilter**
- These fields are lookup keys found in report file, we only expose a subset of these fields suitable for printing purposes

```
$ buildtest report --helpformat
Fields      Description
----------  ------------------------------------------------------------
buildspec   Buildspec file
command     Command executed
compiler    Retrieve compiler used for test (applicable for compiler schema)
endtime     End Time for Test in date format
errfile     Error File
executor    Executor name
hostname    Retrieve hostname of machine where job was submitted from
full_id     Full qualified unique build identifier
id          Unique Build Identifier (abbreviated)
name        Name of test defined in buildspec
outfile     Output file
returncode  Return Code from Test Execution
runtime     Total runtime in seconds
schemafile  Schema file used for validation
starttime   Start Time of test in date format
state       Test State reported by buildtest (PASS/FAIL)
tags        Tag name
testroot    Root of test directory
testpath    Path to test
user        Get user who submitted job
```

```
$ buildtest report --helpfilter
Filter Fields    Description                 Expected Value
---------------  --------------------------  ----------------
buildspec        Filter by buildspec file    FILE
name             Filter by test name         STRING
executor         Filter by executor name     STRING
state            Filter by test state        PASS/FAIL
tags             Filter tests by tag name    STRING
returncode       Filter tests by returncode  INT
```

Inspect a Test

- Buildtest stores all test results in JSON file ($BUILDTEST_ROOT/var/report.json) for retrieval
- The **buildtest inspect** command can retrieve test records from this file.
- We can retrieve all test names and corresponding test IDs using **buildtest inspect list**
- The **buildtest inspect name** can retrieve test records based on test names including all previous runs
- You can pass multiple test names to **buildtest inspect name** command to query multiple records

```
(buildtest) bash-3.2$ buildtest inspect list
+--------------------------+------------------------------------------+-----------------------------------------------------------------------------------------+
| name                     | id                                       | buildspec                                                                               |
+==========================+==========================================+=========================================================================================+
| variables_bash           | c9c42d6f-6f08-456c-a463-5714379dfaf4 | /Users/siddiq90/Documents/GitHubDesktop/buildtest/tutorials/vars.yml                    |
+--------------------------+------------------------------------------+-----------------------------------------------------------------------------------------+
| sleep                    | b0d850d0-04b7-43d1-bd1a-cce0c480a214 | /Users/siddiq90/Documents/GitHubDesktop/buildtest/tutorials/sleep.yml                   |
+--------------------------+------------------------------------------+-----------------------------------------------------------------------------------------+
| exit1_fail               | 996d9773-4523-4cca-bd42-cbc1a4e256ec | /Users/siddiq90/Documents/GitHubDesktop/buildtest/tutorials/pass_returncode.yml         |
+--------------------------+------------------------------------------+-----------------------------------------------------------------------------------------+
| exit1_pass               | 84a5745c-3be4-41c9-864c-30fd11c2ea82 | /Users/siddiq90/Documents/GitHubDesktop/buildtest/tutorials/pass_returncode.yml         |
+--------------------------+------------------------------------------+-----------------------------------------------------------------------------------------+
| returncode_list_mismatch | 6049eea8-4b83-44fc-8b78-d5d3ba90f77f | /Users/siddiq90/Documents/GitHubDesktop/buildtest/tutorials/pass_returncode.yml         |
+--------------------------+------------------------------------------+-----------------------------------------------------------------------------------------+
| returncode_int_match     | 24959464-aa7b-43c7-9db4-0f11ef89b710 | /Users/siddiq90/Documents/GitHubDesktop/buildtest/tutorials/pass_returncode.yml         |
+--------------------------+------------------------------------------+-----------------------------------------------------------------------------------------+
```

Inspect a Test Record

```
(buildtest) bash-3.2$ buildtest inspect name sleep
{
  "sleep": {
    "id": "b0d850d0",
    "full_id": "b0d850d0-04b7-43d1-bd1a-cce0c480a214",
    "description": "sleep 2 seconds",
    "schemafile": "script-v1.0.schema.json",
    "executor": "generic.local.bash",
    "compiler": null,
    "hostname": "DOE-7086392.local",
    "user": "siddiq90",
    "testroot": "/Users/siddiq90/Documents/GitHubDesktop/buildtest/var/tests/generic.local.bash/sleep/sleep/b0d850d0",
    "testpath": "/Users/siddiq90/Documents/GitHubDesktop/buildtest/var/tests/generic.local.bash/sleep/sleep/b0d850d0/sleep.sh",
    "stagedir": "/Users/siddiq90/Documents/GitHubDesktop/buildtest/var/tests/generic.local.bash/sleep/sleep/b0d850d0/stage",
    "command": "sh sleep_build.sh",
    "outfile": "/Users/siddiq90/Documents/GitHubDesktop/buildtest/var/tests/generic.local.bash/sleep/sleep/b0d850d0/sleep.out",
    "errfile": "/Users/siddiq90/Documents/GitHubDesktop/buildtest/var/tests/generic.local.bash/sleep/sleep/b0d850d0/sleep.err",
    "buildspec_content": "version: \"1.0\"\nbuildspecs:\n  sleep:\n    type: script\n    executor: generic.local.bash\n    description: sleep 2 seconds\n
tags: [tutorials]\n    vars:\n      SLEEP_TIME: 2\n    run: sleep $SLEEP_TIME",
    "test_content": "#!/bin/bash \n# Declare shell variables\nSLEEP_TIME=2\n\n\n# Content of run section\nsleep $SLEEP_TIME",
    "buildscript_content": "#!/bin/bash\n\n\n############## START VARIABLE DECLARATION ##########################\nexport BUILDTEST_TEST_NAME=sleep\nexport BUIL
DTEST_TEST_ROOT=/Users/siddiq90/Documents/GitHubDesktop/buildtest/var/tests/generic.local.bash/sleep/sleep/b0d850d0\nexport BUILDTEST_BUILDSPEC_DIR=/Users/si
ddiq90/Documents/GitHubDesktop/buildtest/tutorials\nexport BUILDTEST_STAGE_DIR=/Users/siddiq90/Documents/GitHubDesktop/buildtest/var/tests/generic.local.bash
/sleep/sleep/b0d850d0/stage\nexport BUILDTEST_TEST_ID=b0d850d0-04b7-43d1-bd1a-cce0c480a214\n############## END VARIABLE DECLARATION   ##########################
\n\n\n# source executor startup script\nsource /Users/siddiq90/Documents/GitHubDesktop/buildtest/var/executor/generic.local.bash/before_script.sh\n# Run gene
rated script\n/Users/siddiq90/Documents/GitHubDesktop/buildtest/var/tests/generic.local.bash/sleep/sleep/b0d850d0/stage/sleep.sh\n# Get return code\nreturnco
de=$?\n# Exit with return code\nexit $returncode",
    "logpath": "/var/folders/1m/_jjv09h17k37mkktwnmbkmj0002t_q/T/buildtest_04rwo4dk.log",
    "metrics": {},
    "tags": "tutorials",
    "starttime": "2021/08/16 16:37:35",
    "endtime": "2021/08/16 16:37:38",
    "runtime": "2.085489",
    "state": "PASS",
    "returncode": "0",
    "output": "",
    "error": "",
    "job": {},
    "build_script": "/Users/siddiq90/Documents/GitHubDesktop/buildtest/var/tests/generic.local.bash/sleep/sleep/b0d850d0/sleep_build.sh"
  }
}
```

buildtest inspect query

- The **buildtest inspect query** command can be used to query individual test records.
- By default, it will retrieve the latest run for given test, however one can use **–d** option to retrieve all records or first or last record

```
(buildtest) bash-3.2$ buildtest inspect query python_hello
_____________________________ python_hello (ID: 5b5a4a26-7b1d-4d1d-8223-3e731cb7bcfa) _____________________________
executor:  generic.local.bash
description:  Hello World python
state:  PASS
returncode:  0
runtime:  0.27114
starttime:  2021/08/25 20:44:49
endtime:  2021/08/25 20:44:50
```

```
(buildtest) bash-3.2$ buildtest inspect query -d all  python_hello
_____________________________ python_hello (ID: b74e5320-6e47-429b-9281-445ecc0cb002) _____________________________
executor:  generic.local.bash
description:  Hello World python
state:  PASS
returncode:  0
runtime:  0.40071
starttime:  2021/08/25 20:44:48
endtime:  2021/08/25 20:44:48
_____________________________ python_hello (ID: 5b5a4a26-7b1d-4d1d-8223-3e731cb7bcfa) _____________________________
executor:  generic.local.bash
description:  Hello World python
state:  PASS
returncode:  0
runtime:  0.27114
starttime:  2021/08/25 20:44:49
endtime:  2021/08/25 20:44:50
```

buildtest inspect query

```
(buildtest) bash-3.2$ buildtest inspect query --output  python_hello
_____________________________ python_hello (ID: 5b5a4a26-7b1d-4d1d-8223-3e731cb7bcfa) _____________________________
executor:  generic.local.bash
description:  Hello World python
state:  PASS
returncode:  0
runtime:  0.27114
starttime:  2021/08/25 20:44:49
endtime:  2021/08/25 20:44:50
************************* Start of Output File: /Users/siddiq90/Documents/GitHubDesktop/buildtest/var/tests/generic.local.bash/python-hello/python_hello/5b5a4a26/python_hello.out *************************
Hello World

************************* End of Output File: /Users/siddiq90/Documents/GitHubDesktop/buildtest/var/tests/generic.local.bash/python-hello/python_hello/5b5a4a26/python_hello.out *************************
```

```
(buildtest) bash-3.2$ buildtest inspect query --test --output --error  python_hello
_____________________________ python_hello (ID: 5b5a4a26-7b1d-4d1d-8223-3e731cb7bcfa) _____________________________
executor:  generic.local.bash
description:  Hello World python
state:  PASS
returncode:  0
runtime:  0.27114
starttime:  2021/08/25 20:44:49
endtime:  2021/08/25 20:44:50
************************* Start of Output File: /Users/siddiq90/Documents/GitHubDesktop/buildtest/var/tests/generic.local.bash/python-hello/python_hello/5b5a4a26/python_hello.out *************************
Hello World

************************* End of Output File: /Users/siddiq90/Documents/GitHubDesktop/buildtest/var/tests/generic.local.bash/python-hello/python_hello/5b5a4a26/python_hello.out *************************

************************* Start of Error File:  /Users/siddiq90/Documents/GitHubDesktop/buildtest/var/tests/generic.local.bash/python-hello/python_hello/5b5a4a26/python_hello.err *************************

************************* End of Error File:  /Users/siddiq90/Documents/GitHubDesktop/buildtest/var/tests/generic.local.bash/python-hello/python_hello/5b5a4a26/python_hello.err *************************

************************* Start of Test Path:  /Users/siddiq90/Documents/GitHubDesktop/buildtest/var/tests/generic.local.bash/python-hello/python_hello/5b5a4a26/python_hello.sh *************************
#!/bin/bash
# Content of run section
python hello.py
************************* End of Test Path:  /Users/siddiq90/Documents/GitHubDesktop/buildtest/var/tests/generic.local.bash/python-hello/python_hello/5b5a4a26/python_hello.sh *************************
```

Demo – Buildspec Interface & Query Test Report

Get Path to tests

- The **buildtest path** command can retrieve path to test given a test name. If no options are specified we retrieve the root where test is available.
- You can specify a test ID by specifying name followed by **backslash (/)** and name of test ID if its not specified buildtest will fetch the latest run.

```
(buildtest) bash-3.2$ buildtest path shell_options
/Users/siddiq90/Documents/GitHubDesktop/buildtest/var/tests/generic.local.sh/shell_examples/shell_options/f91f29d6

(buildtest) bash-3.2$ buildtest path shell_options/b89
/Users/siddiq90/Documents/GitHubDesktop/buildtest/var/tests/generic.local.sh/shell_examples/shell_options/b890cd6d

(buildtest) bash-3.2$ cat $(buildtest path -t shell_options)
#!/bin/sh -x
# Content of run section
echo $SHELL
```

Query Previous Builds

- The **buildtest history query** command can be used to query previous builds based on build identifier. Every **buildtest build** command will be stored as a new build identifier

Build Details

Discovered Buildspecs

Test Summary

Builder Details

```
(buildtest) bash-3.2$ buildtest history query 0
{
  "command": "/Users/siddiq90/Documents/GitHubDesktop/buildtest/bin/buildtest build -b /Users/siddiq90/Documents/GitHubDesktop/buildtest/tutorials/vars.yml",
  "user": "siddiq90",
  "hostname": "DOE-7086392.local.dhcp.lbl.gov",
  "platform": "Darwin",
  "date": "2021/08/25 20:43:45",
  "buildtest": "/Users/siddiq90/Documents/GitHubDesktop/buildtest/bin/buildtest",
  "python": "/Users/siddiq90/.local/share/virtualenvs/buildtest-KLOcDrW0/bin/python",
  "python_version": "3.7.3",
  "testdir": "/Users/siddiq90/Documents/GitHubDesktop/buildtest/var/tests",
  "configuration": "/Users/siddiq90/Documents/GitHubDesktop/buildtest/buildtest/settings/config.yml",
  "system": "generic",
  "logpath": "/Users/siddiq90/Documents/GitHubDesktop/buildtest/var/.history/0/buildtest_6cv_jp2b.log",
  "invalid_buildspecs": [],
  "buildspecs": {
    "detected": [
      "/Users/siddiq90/Documents/GitHubDesktop/buildtest/tutorials/vars.yml"
    ],
    "included": [
      "/Users/siddiq90/Documents/GitHubDesktop/buildtest/tutorials/vars.yml"
    ],
    "excluded": []
  },
  "test_summary": {
    "pass": "1",
    "fail": "0",
    "total": "1",
    "pass_rate": "100.000",
    "fail_rate": "0.000"
  },
  "builders": {
    "d1aef7d1-6386-4bb9-b142-aa045724adbc": {
      "name": "variables_bash",
      "buildspec": "/Users/siddiq90/Documents/GitHubDesktop/buildtest/tutorials/vars.yml",
      "tags": [
        "tutorials"
      ],
      "executors": "generic.local.bash",
      "state": "PASS",
      "returncode": 0,
      "runtime": 0.549762,
      "testpath": "/Users/siddiq90/Documents/GitHubDesktop/buildtest/var/tests/generic.local.bash/vars/variables_bash/d1aef7d1/variables_bash.sh",
      "errfile": "/Users/siddiq90/Documents/GitHubDesktop/buildtest/var/tests/generic.local.bash/vars/variables_bash/d1aef7d1/variables_bash.err",
      "outfile": "/Users/siddiq90/Documents/GitHubDesktop/buildtest/var/tests/generic.local.bash/vars/variables_bash/d1aef7d1/variables_bash.out"
    }
  }
}
```

Spack support in buildtest

- Spack support was added recently in buildtest v0.10.0 to write buildspecs using the spack schema.
- Current support includes
  - Installing specs
  - Managing environments (create, activate, remove)
  - **spack test** support
  - Specify scheduler options.
- For more details on spack support see https://buildtest.readthedocs.io/en/devel/buildspecs/spack.html

```yaml
version: "1.0"
buildspecs:
  install_zlib:
    type: spack
    executor: generic.local.sh
    description: "Install zlib"
    tags: [spack]
    spack:
      root: $HOME/spack
      install:
        specs: ['zlib']
```

Use Spack schema

Root of spack

List of specs to install

```
#!/bin/bash
source /Users/siddiq90/spack/share/spack/setup-env.sh
spack install  zlib
```

Activate Spack environment

- The **env** property is used for managing spack environment and maps to **spack env** command
- The **activate** property maps to **spack env activate** used for activating named environment
- **concretize** is a boolean type that will determine if **spack concretize –f** will be injecting in the test.

```yaml
version: "1.0"
buildspecs:
  concretize_m4_in_spack_env:
    type: spack
    executor: generic.local.sh
    description: "Concretize m4 in a spack environment named m4"
    tags: [spack]
    spack:
      root: $HOME/spack
      env:
        specs:
          - 'm4'
        activate:
          name: m4
        concretize: true
```

List of specs to add in environment

Activate spack environment

Spack conretize

```
#!/bin/bash
source /Users/siddiq90/spack/share/spack/setup-env.sh
spack env activate  m4
spack add m4
spack concretize -f
```

```
==> Package m4 was already added to m4
==> Concretized m4
[+]  volmsbn  m4@1.4.19%apple-clang@11.0.3+sigsegv arch=darwin-bigsur-skylake
[+]  bc6kuc4      ^libsigsegv@2.13%apple-clang@11.0.3 arch=darwin-bigsur-skylake
```

Create spack environment

- The **create** is a property under **env** that is used for creating spack environment.
- User is responsible for activate spack environment upon creation.
- The **install** property maps to **spack install** and one can pass options via **option** property

```
==> Found no new compilers
==> Compilers are defined in the following files:
   /Users/siddiq90/.spack/darwin/compilers.yaml
==> Updating view at /Users/siddiq90/spack/var/spack/environments/m4_zlib/.spack-env/view
==> Created environment 'm4_zlib' in /Users/siddiq90/spack/var/spack/environments/m4_zlib
==> You can activate this environment with:
==>   spack env activate m4_zlib
==> Adding m4 to environment m4_zlib
==> Adding zlib to environment m4_zlib
==> Concretized m4
[+]  volmsbn  m4@1.4.19%apple-clang@11.0.3+sigsegv arch=darwin-bigsur-skylake
[+]  bc6kuc4      ^libsigsegv@2.13%apple-clang@11.0.3 arch=darwin-bigsur-skylake
==> Concretized zlib
 -   2hw3hzd  zlib@1.2.11%apple-clang@11.0.3+optimize+pic+shared arch=darwin-bigsur-skylake
==> Updating view at /Users/siddiq90/spack/var/spack/environments/m4_zlib/.spack-env/view
==> Installing environment m4_zlib
==> Installing zlib-1.2.11-2hw3hzdfy7e2ndzojgqoq472m5flsloj
==> No binary for zlib-1.2.11-2hw3hzdfy7e2ndzojgqoq472m5flsloj found: installing from source
==> Fetching https://mirror.spack.io/_source-cache/archive/c3/c3e5e9fdd5004dcb542feda5ee4f0ff0
==> No patches needed for zlib
==> zlib: Executing phase: 'install'
==> zlib: Successfully installed zlib-1.2.11-2hw3hzdfy7e2ndzojgqoq472m5flsloj
  Fetch: 0.84s.  Build: 6.98s.  Total: 7.82s.
[+] /Users/siddiq90/spack/opt/spack/darwin-bigsur-skylake/apple-clang-11.0.3/zlib-1.2.11-2hw3h
==> Updating view at /Users/siddiq90/spack/var/spack/environments/m4_zlib/.spack-env/view
```

```
#!/bin/bash
source /Users/siddiq90/spack/share/spack/setup-env.sh
spack compiler find
spack env create  m4_zlib
spack env activate  m4_zlib
spack add m4
spack add zlib
spack concretize -f
spack install --keep-prefix
```

```
version: "1.0"
buildspecs:
  install_m4_zlib_in_spack_env:
    type: spack
    executor: generic.local.sh
    description: "Install m4 and zlib in a spack environment named m4_zlib"
    tags: [spack]
    spack:
      root: $HOME/spack
      compiler_find: true
      env:
        create:
          name: 'm4_zlib'
        specs:
          - 'm4'
          - 'zlib'
        activate:
          name: m4_zlib
        concretize: true
      install:
        option: '--keep-prefix'
```

Find spack compilers

Create spack environment

Pass options to **spack install**

Creating spack environment via spack.yaml

- We can create spack environment based on **spack.yaml** which can be specified via **manifest** property which expects path to spack.yaml file.

```yaml
version: "1.0"
buildspecs:
  spack_env_create_from_manifest:
    type: spack
    executor: generic.local.sh
    description: "Create spack environment from spack.yaml"
    tags: [spack]
    spack:
      root: $HOME/spack
      env:
        create:
          name: 'manifest_example'
          manifest: "$BUILDTEST_ROOT/tutorials/spack/example/spack.yaml"
        activate:
          name: 'manifest_example'
        concretize: true
```

Remove spack environment

- Buildtest provides two methods for removing spack environment, one is via **rm** property which gives user control over how to remove spack environment. The alternative is let buildtest automatically remove environment which can be specified via **remove_environment** which expects a boolean.
- The **remove_environment** is a property under **create** while **rm** is property under **env** which maps to **spack env rm**.
- The **remove_environment** will remove environment based on **name** property

```
#!/bin/bash
source /Users/siddiq90/spack/share/spack/setup-env.sh
spack env rm -y remove_environment
spack env create  remove_environment
spack env activate  remove_environment
spack add bzip2
spack concretize -f
```

```
#!/bin/bash
source /Users/siddiq90/spack/share/spack/setup-env.sh
spack env rm -y dummy
spack env create  dummy
spack env activate  dummy
spack add bzip2
spack concretize -f
```

```
version: "1.0"
buildspecs:
  remove_environment_automatically:
    type: spack
    executor: generic.local.sh
    description: "remove spack environment automatically before creating a new environment"
    tags: [spack]
    spack:
      root: $HOME/spack
      env:
        create:
          remove_environment: true
          name: remove_environment
        activate:
          name: remove_environment
        specs:
          - 'bzip2'
        concretize: true

  remove_environment_explicit:
    type: spack
    executor: generic.local.sh
    description: "remove spack environment explicitly using the 'rm' property"
    tags: [spack]
    spack:
      root: $HOME/spack
      env:
        rm:
          name: dummy
        create:
          name: dummy
        activate:
          name: dummy
        specs:
          - 'bzip2'
        concretize: true
```

Running test via **spack test**

- The **test** property maps to **spack test** command and **run** expects a list of specs to run that is defined via **specs**.
- Buildtest will write one line per **spack test run** and create an alias for each spec so one can retrieve the result via suite name
- **pre_cmds** are list of commands run before sourcing spack
- **post_cmds** are list of commands run after spack

```
  Fetch: 1.32s.  Build: 52.35s.  Total: 53.67s.
[+] /private/tmp/spack/opt/spack/darwin-bigsur-skylake/apple-clang-11.0.3/diffutils-3.7-3dfrh6
==> Installing bzip2-1.0.8-avjwvsoaivuflugopwk4ap7rffhejxzu
==> No binary for bzip2-1.0.8-avjwvsoaivuflugopwk4ap7rffhejxzu found: installing from source
==> Fetching https://mirror.spack.io/_source-cache/archive/ab/ab5a03176ee106d3f0fa90e381da478d
==> Ran patch() for bzip2
==> bzip2: Executing phase: 'install'
==> bzip2: Successfully installed bzip2-1.0.8-avjwvsoaivuflugopwk4ap7rffhejxzu
  Fetch: 1.42s.  Build: 1.84s.  Total: 3.26s.
[+] /private/tmp/spack/opt/spack/darwin-bigsur-skylake/apple-clang-11.0.3/bzip2-1.0.8-avjwvsoa
==> Spack test bzip2
==> Testing package bzip2-1.0.8-avjwvso
==> Results for test suite 'bzip2':
==>   bzip2-1.0.8-avjwvso PASSED
-- darwin-bigsur-skylake / apple-clang@11.0.3 --------------------
bzip2@1.0.8
diffutils@3.7
libiconv@1.16
```

```yaml
version: "1.0"
buildspecs:
  spack_test:
    type: spack
    executor: generic.local.sh
    description: "Install bzip2 and run spack test and report results"
    tags: [spack]
    pre_cmds: |
      cd /tmp
      git clone https://github.com/spack/spack
    spack:
      root: /tmp/spack
      verify_spack: false
      install:
        specs: ['bzip2']
      test:
        run:
          specs: ['bzip2']
        results:
          suite: ['bzip2']

    post_cmds: |
      spack find
      rm -rf $SPACK_ROOT
```

List of specs to for **spack test run**

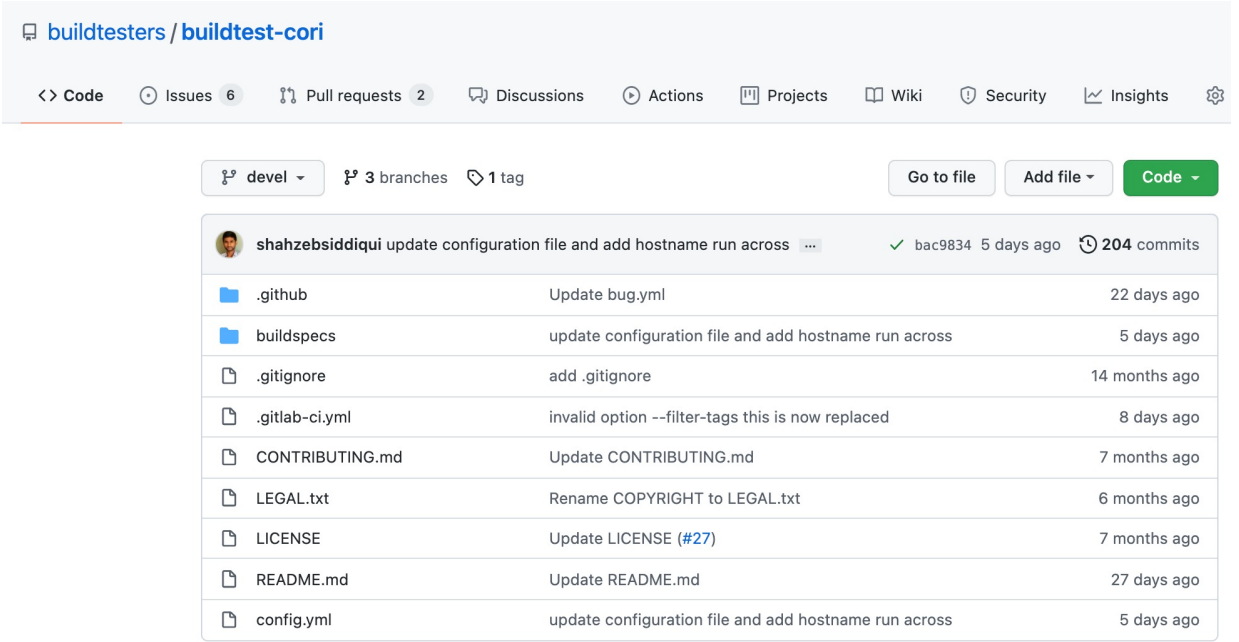
Get test results via **spack test results**

```
#!/bin/bash


######## START OF PRE COMMANDS ########
cd /tmp
git clone https://github.com/spack/spack spack

######## END OF PRE COMMANDS   ########


source /private/tmp/spack-test-no-env/share/spack/setup-env.sh
spack install  bzip2
spack test run  --alias bzip2 bzip2
spack test results  bzip2


######## START OF POST COMMANDS ########
spack find
rm -rf $SPACK_ROOT
######## END OF POST COMMANDS   ########
```

Cori Testsuite

Cori Testsuite

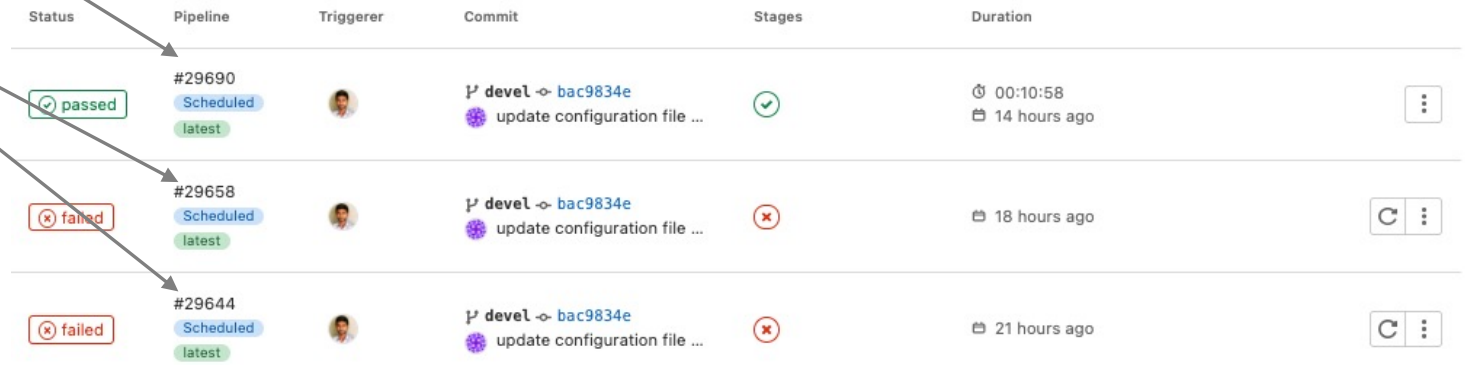
- The Cori Testsuite (https://github.com/buildtesters/buildtest-cori) is buildtest tests that run for Cori and Perlmutter system.
- There is a GitHub-GitLab CI/CD workflow to trigger pipeline at NERSC GitLab server: https://software.nersc.gov
- Test results are pushed to CDASH at https://my.cdash.org/index.php?project=buildtest-cori
- Test are run with a single user name **e4s**



Scheduled Pipelines

- We have a mirror setup at https://software.nersc.gov/siddiq90/buildtest-cori
- Currently, we have two scheduled pipelines for daily system check and E4S tests

DAILYCHECK=TRUE




```
scheduled_system_check:
  stage: test
  only:
    refs:
      - schedules
    variables:
      - $DAILYCHECK == "True"
  tags: [c_e4s_cori01]
  before_script:
    - *setup-buildtest
  script:
    - buildtest -c $CI_PROJECT_DIR/config.yml buildspec find --rebuild --root $CI_PROJECT_DIR/buildspecs
    - buildtest -c $CI_PROJECT_DIR/config.yml build --tags daily --filter tags=daily -r $BUILDTEST_ROOT/report.json  --testdir $CFS/m3503/buildtest/runs/$CI_JOB_NAME/$(date +%F)
    - buildtest report --filter state=FAIL -r $BUILDTEST_ROOT/report.json
    - mkdir -p $CI_PROJECT_DIR/.artifacts
    - cp $BUILDTEST_ROOT/{buildtest.log,report.json} $CI_PROJECT_DIR/.artifacts
    - buildtest -c $CI_PROJECT_DIR/config.yml cdash upload daily_check -r $BUILDTEST_ROOT/report.json
    - source deactivate
  after_script:
    - *cleanup-buildtest

  artifacts:
    paths:
      - $CI_PROJECT_DIR/.artifacts
```

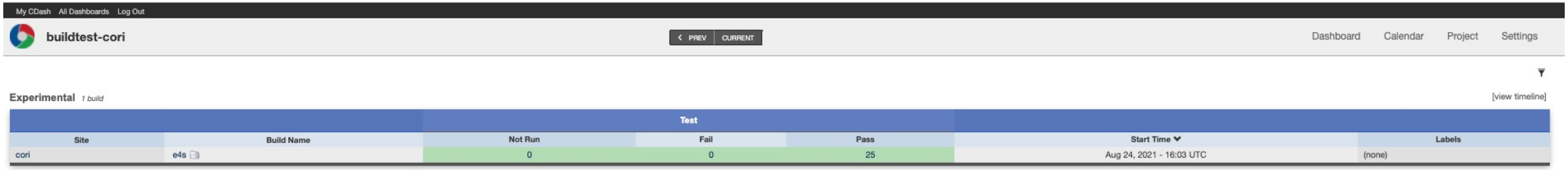
Gitlab Job Output

```
+----------------------+
| Stage: Test Summary  |
+----------------------+

 name                        | id       | executor                 | status   |   returncode |   runtime
-----------------------------+----------+--------------------------+----------+--------------+-----------
 filesystem_benchmark_creates | 2d2bef91 | cori.local.bash          | PASS     |            0 |  11.3265
 stream_uniprocess_c          | 4cc7d4ff | cori.local.bash          | PASS     |            0 |   1.63912
 stream_openmp_c              | c927fecb | cori.local.bash          | PASS     |            0 |   4.45247
 filesystem_benchmark_ls      | 79f36cf2 | cori.local.bash          | PASS     |            0 |  58.2977
 mkl_intel_threaded_dgemm     | 85189a06 | cori.slurm.haswell_debug | PASS     |            0 | 283.427
 mkl_gnu_threaded             | 4f8a1fef | cori.slurm.haswell_debug | PASS     |            0 | 139.516
 libsci_gnu_dgemm             | 65784df9 | cori.slurm.haswell_debug | PASS     |            0 | 333.814
 mkl_intel_sequential_dgemm   | 74f44c66 | cori.slurm.haswell_debug | PASS     |            0 | 582.243
Passed Tests: 8/8 Percentage: 100.000%
Failed Tests: 0/8 Percentage: 0.000%
Writing Logfile to: /tmp/buildtest_cmwvuruk.log
A copy of logfile can be found at $BUILDTEST_ROOT/buildtest.log -  /global/homes/e/e4s/builds/5s9RF_SV/0/siddiq90/buildtest-cori/buildtest/buildtest.log
$ buildtest report --filter state=FAIL -r $BUILDTEST_ROOT/report.json
Reading report file: /global/u1/e/e4s/builds/5s9RF_SV/0/siddiq90/buildtest-cori/buildtest/report.json
+--------+------+---------+--------------+-------------+-----------+-----------+--------+-------------+
| name   | id   | state   | returncode   | starttime   | endtime   | runtime   | tags   | buildspec   |
+========+======+=========+==============+=============+===========+===========+========+=============+
+--------+------+---------+--------------+-------------+-----------+-----------+--------+-------------+
$ mkdir -p $CI_PROJECT_DIR/.artifacts
$ cp $BUILDTEST_ROOT/{buildtest.log,report.json} $CI_PROJECT_DIR/.artifacts
$ buildtest -c $CI_PROJECT_DIR/config.yml cdash upload benchmark -r $BUILDTEST_ROOT/report.json
Reading configuration file:  /global/u1/e/e4s/builds/5s9RF_SV/0/siddiq90/buildtest-cori/config.yml
Reading report file:  /global/u1/e/e4s/builds/5s9RF_SV/0/siddiq90/buildtest-cori/buildtest/report.json
build name:  benchmark
site:  cori
stamp:  20210901-1704-Experimental
MD5SUM: 1f691c22fbeac9f21c1271c59df2e883
PUT STATUS: 200
You can view the results at: https://my.cdash.org/viewTest.php?buildid=2060659
```

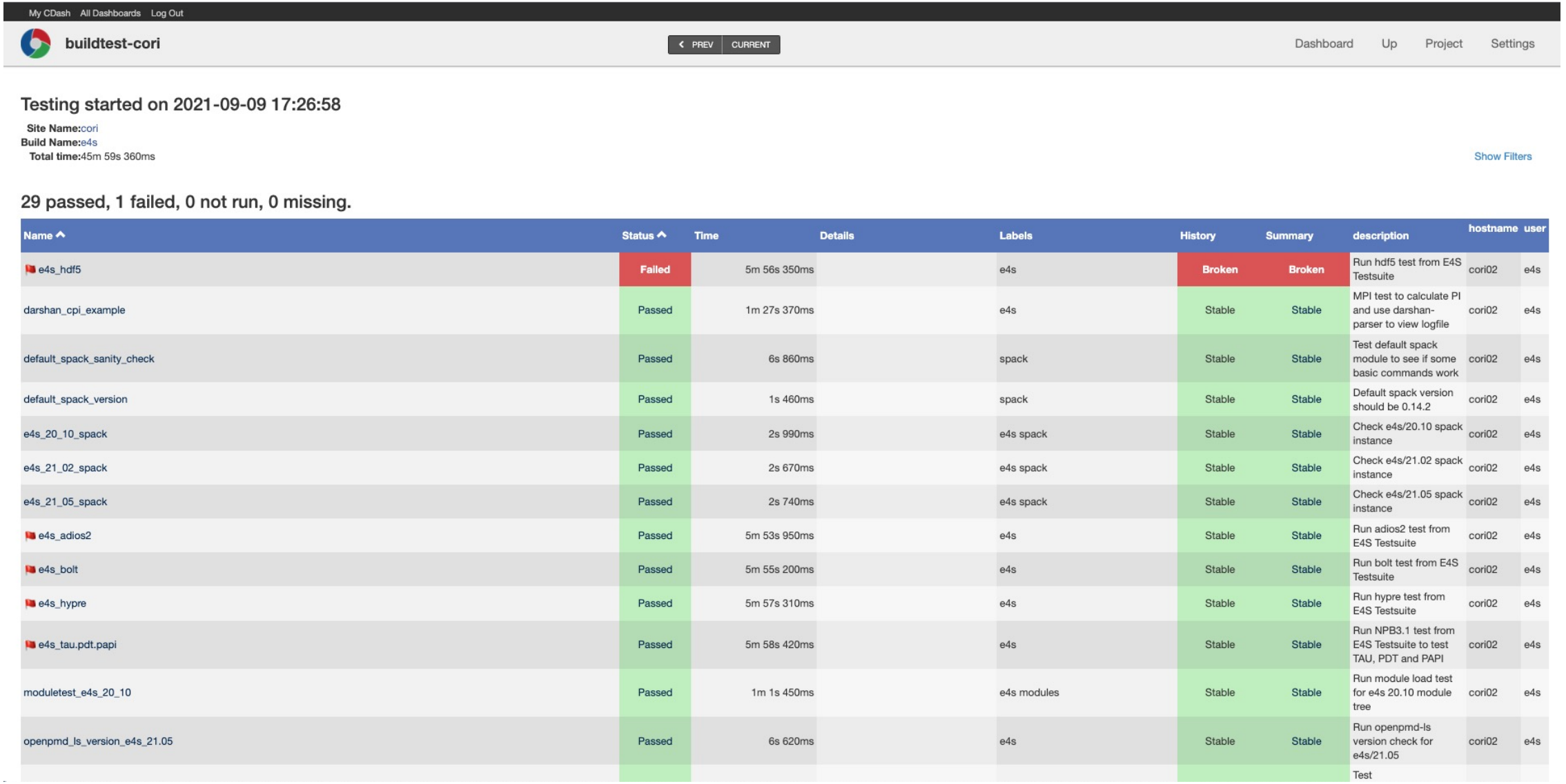
CDASH Results

- We push test results to public CDASH server: https://my.cdash.org/index.php?project=buildtest-cori for both scheduled pipelines.
- The build names correspond to GitLab job
- The **buildtest cdash upload** command can push results to CDASH given a report file. The report file can be passed via **–r** option

```
$ buildtest -c $CI_PROJECT_DIR/config.yml cdash upload e4s -r $BUILDTEST_ROOT/report.json
Reading configuration file:  /global/u1/e/e4s/builds/5s9RF_SV/0/siddiq90/buildtest-cori/config.yml
Reading report file:  /global/u1/e/e4s/builds/5s9RF_SV/0/siddiq90/buildtest-cori/buildtest/report.json
build name:  e4s
site:  cori
stamp:  20210824-0903-Experimental
MD5SUM: 83db323b583b8e0f0f1e170929824f33
PUT STATUS: 200
You can view the results at: https://my.cdash.org/viewTest.php?buildid=2056685
```



CDASH Results



My CDash All Dashboards Log Out

buildtest-cori

PREV CURRENT

Dashboard Up Project Settings

Testing started on 2021-09-09 17:26:58

Site Name:cori
Build Name:e4s
Total time:45m 59s 360ms

Show Filters

29 passed, 1 failed, 0 not run, 0 missing.

Name	Status	Time	Details	Labels	History	Summary	description	hostname	user
e4s_hdf5	Failed	5m 56s 350ms		e4s	Broken	Broken	Run hdf5 test from E4S Testsuite	cori02	e4s
darshan_cpi_example	Passed	1m 27s 370ms		e4s	Stable	Stable	MPI test to calculate PI and use darshan-parser to view logfile	cori02	e4s
default_spack_sanity_check	Passed	6s 860ms		spack	Stable	Stable	Test default spack module to see if some basic commands work	cori02	e4s
default_spack_version	Passed	1s 460ms		spack	Stable	Stable	Default spack version should be 0.14.2	cori02	e4s
e4s_20_10_spack	Passed	2s 990ms		e4s spack	Stable	Stable	Check e4s/20.10 spack instance	cori02	e4s
e4s_21_02_spack	Passed	2s 670ms		e4s spack	Stable	Stable	Check e4s/21.02 spack instance	cori02	e4s
e4s_21_05_spack	Passed	2s 740ms		e4s spack	Stable	Stable	Check e4s/21.05 spack instance	cori02	e4s
e4s_adios2	Passed	5m 53s 950ms		e4s	Stable	Stable	Run adios2 test from E4S Testsuite	cori02	e4s
e4s_bolt	Passed	5m 55s 200ms		e4s	Stable	Stable	Run bolt test from E4S Testsuite	cori02	e4s
e4s_hypre	Passed	5m 57s 310ms		e4s	Stable	Stable	Run hypre test from E4S Testsuite	cori02	e4s
e4s_tau.pdt.papi	Passed	5m 58s 420ms		e4s	Stable	Stable	Run NPB3.1 test from E4S Testsuite to test TAU, PDT and PAPI	cori02	e4s
moduletest_e4s_20_10	Passed	1m 1s 450ms		e4s modules	Stable	Stable	Run module load test for e4s 20.10 module tree	cori02	e4s
openpmd_ls_version_e4s_21.05	Passed	6s 620ms		e4s	Stable	Stable	Run openpmd-ls version check for e4s/21.05	cori02	e4s
							Test		

E4S Tests on Cori

Cori E4S Testing Strategy

- We are testing the facility deployed e4s stacks (e4s/21.05, e4s/21.02, e4s/20.10). Typically one has to load one of the e4s modules **module load e4s/21.05** and run **module load** or **spack load** to load the software before running the test.
- Please see https://docs.nersc.gov/applications/e4s/ for more details regarding our facility deployment of e4s
- We leverage **spack test** and E4S testsuite to test the e4s stack and sometimes we develop tests that are site specific.
- E4S tests are available at https://github.com/buildtesters/buildtest-cori/tree/devel/buildspecs/e4s
- We run all e4s tests using the **e4s** tags: **buildtest build –tags e4s**

```
version: "1.0"
buildspecs:
  spack_test_upcxx_e4s_21.05:
    type: spack
    executor: cori.local.sh
    description: "Test upcxx@2021.03.0 for e4s/21.05 test via spack test"
    tags: e4s
    pre_cmds: |
      module load e4s/21.05
      module swap intel intel/19.1.3.304
    spack:
      root: /global/common/software/spackecp/e4s-21.05/spack/
      verify_spack: false
      test:
        run:
          specs: ['upcxx@2021.03.0%intel']
        results:
          option: '-l'
          specs: ['upcxx@2021.03.0%intel']
maintainers:
- "shahzebsiddiqui"
- "PHHargrove"
- "bonachea"
```

Spack Test Example - Gasnet

```yaml
version: "1.0"
buildspecs:
  spack_test_gasnet_e4s_21.05:
    type: spack
    executor: cori.local.sh
    description: "Test gasnet@2021.3.0%intel with e4s/21.05 via spack test"
    tags: e4s
    pre_cmds: module load e4s/21.05
    spack:
      root: /global/common/software/spackecp/e4s-21.05/spack/
      verify_spack: false
      test:
        run:
          specs: ['gasnet@2021.3.0%intel']
        results:
          option: '-l'
          specs: ['gasnet@2021.3.0%intel']
maintainers:
  - "shahzebsiddiqui"
  - "PHHargrove"
  - "bonachea"
```

Test Content

```
#!/bin/bash


######## START OF PRE COMMANDS ########
module load e4s/21.05
######## END OF PRE COMMANDS   ########


source /global/common/software/spackecp/e4s-21.05/spack/share/spack/setup-env.sh
spack test run  --alias gasnet@2021.3.0%intel gasnet@2021.3.0%intel
spack test results -l -- gasnet@2021.3.0%intel
```

Test output

```
==> Spack test gasnet@2021.3.0%intel
==> Testing package gasnet-2021.3.0-sbw7ukx
==> Results for test suite 'gasnet@2021.3.0%intel', spec matching 'gasnet@2021.3.0%intel':
==>   gasnet-2021.3.0-sbw7ukx PASSED
==> Testing package gasnet-2021.3.0-sbw7ukx
==> [2021-09-07-09:08:33.032522] Running testtools
==> [2021-09-07-09:08:33.033342] '/global/common/software/spackecp/e4s-21.05/software/cray-cn
=====> testtools config=RELEASE=2021.3.0,SPEC=1.17,PTR=64bit,nodebug,PAR,timers_native,membar
hostname is: cori09 (pid=9029)
testtools running...
Running testtools with 100 iterations and 10 threads
System page size is 2^12 == 4096
CPU count estimated to be: 64
Cache line size estimated to be: 64
Physical memory size estimated to be: 503 GB
A: Testing high-performance timers and sleep...
B: Testing zero-byte counting...
C: Testing local membars...
D: Testing local write membars...
E: Testing local read membars...
F: Testing threadkey (sequential)...
G: Testing atomic ops (sequential)...
H: Testing client-provided backtrace code...
Invoking USER for backtrace...
Spawning pthreads...
I: parallel atomic-op barrier test...
J: parallel threadkey test...
K: parallel atomic-op pounding test...
L: parallel dec-test pounding test...
M: parallel word-tearing test...
N: parallel membar test...
O: parallel compare-and-swap test...
P: parallel swap test...
Q: parallel add test...
R: parallel atomic-op fence test...
Done.
```

https://my.cdash.org/test/40848138

OpenPMD Test

Test: openpmd_ls_version_e4s_21.05 **(Passed)**
Build: e4s (cori) on 2021-09-09 17:19:52
Labels: e4s

Test ID	e663fdcf-aabb-4b2b-aa91-fad28f47ff11
Return Code	0
user	e4s
hostname	cori02
description	Run openpmd-ls version check for e4s/21.05
command	sh openpmd_ls_version_e4s_21.05_build.sh
executor	cori.local.bash
tags	e4s
testroot	/global/cfs/cdirs/m3503/buildtest/runs/e4s_tests/2021-09-09/cori.local.bash/openpmd-ls/openpmd_ls_version_e4s_21.05/e663fdcf
stagedir	/global/cfs/cdirs/m3503/buildtest/runs/e4s_tests/2021-09-09/cori.local.bash/openpmd-ls/openpmd_ls_version_e4s_21.05/e663fdcf/stage
build_script	/global/cfs/cdirs/m3503/buildtest/runs/e4s_tests/2021-09-09/cori.local.bash/openpmd-ls/openpmd_ls_version_e4s_21.05/e663fdcf/openpmd_ls_version_e4s_21.05_build.sh
testpath	/global/cfs/cdirs/m3503/buildtest/runs/e4s_tests/2021-09-09/cori.local.bash/openpmd-ls/openpmd_ls_version_e4s_21.05/e663fdcf/openpmd_ls_version_e4s_21.05.sh
outfile	/global/cfs/cdirs/m3503/buildtest/runs/e4s_tests/2021-09-09/cori.local.bash/openpmd-ls/openpmd_ls_version_e4s_21.05/e663fdcf/openpmd_ls_version_e4s_21.05.out
errfile	/global/cfs/cdirs/m3503/buildtest/runs/e4s_tests/2021-09-09/cori.local.bash/openpmd-ls/openpmd_ls_version_e4s_21.05/e663fdcf/openpmd_ls_version_e4s_21.05.err
starttime	2021/09/09 10:20:54
endtime	2021/09/09 10:21:00
logpath	/tmp/buildtest_01pr0g94.log
compiler	
schemafile	script-v1.0.schema.json

View GitLab CI results

Show Command Line
Display graphs: Select...

Test output

```
openpmd-ls (openPMD-api) 0.13.4
Copyright 2017-2020 openPMD contributors
Authors: Axel Huebl et al.
License: LGPLv3+
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.
```

```yaml
version: "1.0"
buildspecs:
  openpmd_ls_version_e4s_21.05:
    type: script
    executor: cori.local.bash
    description: Run openpmd-ls version check for e4s/21.05
    tags: e4s
    run: |
      module load e4s/21.05
      spack load openpmd-api@0.13.4
      openpmd-ls --version
    status:
      regex:
        stream: stdout
        exp: '(openpmd-ls) \(openPMD-api\) 0.13.4'
maintainers:
  - ax3l
  - shahzebsiddiqui
```

https://my.cdash.org/test/40848126

UPC Test

```yaml
version: "1.0"
buildspecs:
  upc_hello_e4s:
    type: compiler
    executor: cori.slurm.haswell_debug
    description: Hello world in upc for e4s/20.10
    source: src/hello_upcxx.cpp
    tags: [compile, e4s]
    compilers:
      name: ["^builtin_gcc$"]
      default:
        gcc:
          sbatch: ["-t 10", "-N 1"]
          cxx: "upcxx"
          cxxflags: "-L/opt/cray/pe/pmi/default/lib64 -L/opt/cray/ugni/default/lib64 -L/opt/cray/udreg/default/lib64 -L/opt/cray/xpmem/default/lib64 -g --network=aries"
          pre_build: |
            module swap intel intel/19.1.2.254
            module load e4s/20.10
            spack load upcxx@2020.3.0
          run: "upcxx-run -N 1 -n 8 $_EXEC"
maintainers:
  - PHHargrove
  - bonachea
```

Test output

```
Hello from 0 of 8
Hello from 1 of 8
Hello from 2 of 8
Hello from 3 of 8
Hello from 4 of 8
Hello from 5 of 8
Hello from 6 of 8
Hello from 7 of 8
```

Test Content

```
#!/bin/bash
####### START OF SCHEDULER DIRECTIVES #######
#SBATCH -t 10
#SBATCH -N 1
#SBATCH --job-name=upc_hello_e4s
#SBATCH --output=upc_hello_e4s.out
#SBATCH --error=upc_hello_e4s.err
####### END OF SCHEDULER DIRECTIVES   #######


# name of executable
_EXEC=hello_upcxx.cpp.exe
### START OF PRE BUILD SECTION ###
module swap intel intel/19.1.2.254
module load e4s/20.10
spack load upcxx@2020.3.0

### END OF PRE BUILD SECTION   ###


# Compilation Line
upcxx -L/opt/cray/pe/pmi/default/lib64 -L/opt/cray/ugni/default/lib64 -L/opt/cray/udreg/default/lib64 -L/opt/cray/xpmem/default/lib64 -g --network=aries -o $_EXEC /global/u1/e/e4s/builds/5s9RF_SV/0/siddiq90/buildtest-cori/buildspecs/apps/upc/src/hello_upcxx.cpp


# Run executable
upcxx-run -N 1 -n 8 $_EXEC
```

https://my.cdash.org/test/40848145

E4S Testsuite Example – ADIOS2

Test: e4s_adios2 **(Passed)**
Build: e4s (cori) on 2021-09-09 17:19:52
Labels: e4s

Test ID	84225b0c-bb0c-42ff-be56-67dbebf09e67
Return Code	0
user	e4s
hostname	cori02
description	Run adios2 test from E4S Testsuite
command	sh e4s_adios2_build.sh
executor	cori.slurm.haswell_premium
tags	e4s
testroot	/global/cfs/cdirs/m3503/buildtest/runs/e4s_tests/2021-09-09/cori.slurm.haswell_premium/e4s_21.05/e4s_adios2/84225b0c
stagedir	/global/cfs/cdirs/m3503/buildtest/runs/e4s_tests/2021-09-09/cori.slurm.haswell_premium/e4s_21.05/e4s_adios2/84225b0c/stage
build_script	/global/cfs/cdirs/m3503/buildtest/runs/e4s_tests/2021-09-09/cori.slurm.haswell_premium/e4s_21.05/e4s_adios2/84225b0c/e4s_adios2_build.sh
testpath	/global/cfs/cdirs/m3503/buildtest/runs/e4s_tests/2021-09-09/cori.slurm.haswell_premium/e4s_21.05/e4s_adios2/84225b0c/e4s_adios2.sh
outfile	/global/cfs/cdirs/m3503/buildtest/runs/e4s_tests/2021-09-09/cori.slurm.haswell_premium/e4s_21.05/e4s_adios2/84225b0c/e4s_adios2.out
errfile	/global/cfs/cdirs/m3503/buildtest/runs/e4s_tests/2021-09-09/cori.slurm.haswell_premium/e4s_21.05/e4s_adios2/84225b0c/e4s_adios2.err
starttime	2021/09/09 10:20:56
endtime	2021/09/09 10:26:50
logpath	/tmp/buildtest_01pr0g94.log
compiler	
schemafile	script-v1.0.schema.json

View GitLab CI results

```yaml
version: "1.0"
buildspecs:
  e4s_adios2:
    type: script
    executor: cori.slurm.haswell_premium
    description: Run adios2 test from E4S Testsuite
    tags: [e4s]
    sbatch: ["-t 30", "-N 1"]
    run: |
      module swap intel intel/19.1.3.304
      module load e4s/21.05
      git clone https://github.com/E4S-Project/testsuite
      cd testsuite
      source ./setup.sh
      sh test-all.sh --color-off validation_tests/adios2 --print-logs --settings settings.cori.sh
```

```
---CLEANUP LOG---
rm -f *.o hello-world
Compiling /global/cfs/cdirs/m3503/buildtest/runs/e4s_tests/2021-09-09/cori.slurm.haswell_premium/e4s_21.05/e4s_adios2/84225b0c/stage/testsuite/validation_tests/adios2
---COMPILE LOG---
Skipping load: Environment already setup
+ export ADIOS2_LIB_PATH=/global/common/software/spackecp/e4s-21.05/software/cray-cnl7-haswell/intel-19.1.3.304/adios2-2.7.1-l6vlr4kor7tdkynmhgdai5giwrwlode7/lib
+ ADIOS2_LIB_PATH=/global/common/software/spackecp/e4s-21.05/software/cray-cnl7-haswell/intel-19.1.3.304/adios2-2.7.1-l6vlr4kor7tdkynmhgdai5giwrwlode7/lib
+ [[ ! -d /global/common/software/spackecp/e4s-21.05/software/cray-cnl7-haswell/intel-19.1.3.304/adios2-2.7.1-l6vlr4kor7tdkynmhgdai5giwrwlode7/lib ]]
+ export ADIOS2_LIB_PATH=/global/common/software/spackecp/e4s-21.05/software/cray-cnl7-haswell/intel-19.1.3.304/adios2-2.7.1-l6vlr4kor7tdkynmhgdai5giwrwlode7/lib64
+ ADIOS2_LIB_PATH=/global/common/software/spackecp/e4s-21.05/software/cray-cnl7-haswell/intel-19.1.3.304/adios2-2.7.1-l6vlr4kor7tdkynmhgdai5giwrwlode7/lib64
+ make
CC  -I/global/common/software/spackecp/e4s-21.05/software/cray-cnl7-haswell/intel-19.1.3.304/adios2-2.7.1-l6vlr4kor7tdkynmhgdai5giwrwlode7/include -Wall   -c -o hello-
CC  -o hello-world hello-world.o -L/global/common/software/spackecp/e4s-21.05/software/cray-cnl7-haswell/intel-19.1.3.304/adios2-2.7.1-l6vlr4kor7tdkynmhgdai5giwrwlode7
Running /global/cfs/cdirs/m3503/buildtest/runs/e4s_tests/2021-09-09/cori.slurm.haswell_premium/e4s_21.05/e4s_adios2/84225b0c/stage/testsuite/validation_tests/adios2
Skipping load: Environment already setup
Hello World from ADIOS2
Hello World from ADIOS2
Hello World from ADIOS2
Hello World from ADIOS2
Hello World from ADIOS2
Hello World from ADIOS2
Hello World from ADIOS2
Hello World from ADIOS2
Success
```

https://my.cdash.org/test/40848140

Current Issues

- superlu@5.2.1 (https://github.com/buildtesters/buildtest-cori/issues/70) – Permission Error writing make.inc file in install directory
- hypre@2.20 (https://github.com/buildtesters/buildtest-cori/issues/69) – Can't find mpicc
- raja@0.13.0 (https://github.com/buildtesters/buildtest-cori/issues/68 and https://github.com/spack/spack/issues/25047) – Unable to find shared library libRAJA.so

```
==> Error: PermissionError: [Errno 13] Permission denied: '/global/common/software/spackecp/e4s-21.05/sof
/global/common/software/spackecp/e4s-21.05/spack/var/spack/repos/builtin/packages/superlu/package.py:163,
       160
       161        # Write configuration options to make.inc file
       162        make_file_inc = join_path(self.install_test_root, self.make_hdr_file)
  >>   163        with open(make_file_inc, 'w') as inc:
       164            for option in config_args:
       165                inc.write('{0}\n'.format(option))
       166
```

```
==> Spack test hypre%intel~openmp
==> Testing package hypre-2.20.0-ngvypfo
==> Spack test hypre%intel+openmp
==> Testing package hypre-2.20.0-awsizb4
==> Results for test suite 'hypre%intel~openmp', spec matching 'hypre%intel~openmp':
==>   hypre-2.20.0-ngvypfo FAILED
==> Testing package hypre-2.20.0-ngvypfo
==> [2021-08-31-09:04:49.445446] 'make' '-j16' 'HYPRE_DIR=/global/common/software/spackecp
mpicc -g -Wall -I/global/common/software/spackecp/e4s-21.02/software/cray-cnl7-haswell/int
make: mpicc: Command not found
make: *** [Makefile:54: ex5big.o] Error 127
```

```
==> Error: TestFailure: 7 tests failed.
Command exited with status 127:
    './ex5_line-of-sight_solution'
./ex5_line-of-sight_solution: error while loading shared libraries: libRAJA.so: cannot open shared object file:
1 error found in test log:
     1    ==> Testing package raja-0.13.0-xjqerqf
     2    ==> [2021-08-31-09:07:09.369113] test: checking output of ex5_line-of
          -sight_solution for ['RAJA sequential', 'RAJA OpenMP', 'result -- PAS
          S']
     3    ==> [2021-08-31-09:07:09.369798] './ex5_line-of-sight_solution'
     4    ./ex5_line-of-sight_solution: error while loading shared libraries: l
          ibRAJA.so: cannot open shared object file: No such file or directory
     5    FAILED: Command exited with status 127:
     6        './ex5_line-of-sight_solution'
  >> 7    ./ex5_line-of-sight_solution: error while loading shared libraries: l
          ibRAJA.so: cannot open shared object file: No such file or directory
     8
     9      File "/global/common/software/spackecp/e4s-21.02/spack/bin/spack",
          line 71, in <module>
```

Available E4S Tests in buildtest Cori

```
(buildtest) siddiq90@cori12> buildtest -c config.yml buildspec find --filter tags=e4s --format name,tags,description
```

name	tags	description
e4s_adios2	e4s	Run adios2 test from E4S Testsuite
e4s_bolt	e4s	Run bolt test from E4S Testsuite
e4s_hdf5	e4s	Run hdf5 test from E4S Testsuite
e4s_hypre	e4s	Run hypre test from E4S Testsuite
e4s_NPB3.1	e4s	Run NPB3.1 test from E4S Testsuite to test TAU, PDT and PAPI
e4s_adios2	e4s	Run adios2 test from E4S Testsuite
e4s_bolt	e4s	Run bolt test from E4S Testsuite
e4s_hdf5	e4s	Run hdf5 test from E4S Testsuite
e4s_hypre	e4s	Run hypre test from E4S Testsuite
e4s_tau.pdt.papi	e4s	Run NPB3.1 test from E4S Testsuite to test TAU, PDT and PAPI
spack_test_hdf5_e4s_21.02	e4s	Test hdf5@1.10.7%gcc and hdf5@1.10.7%intel from e4s/21.02 using spack test
spack_test_hdf5_e4s_21.05	e4s	Test hdf5@1.8.22%intel and hdf5@1.10.7%intel from e4s/21.05 using spack test
spack_test_umpire_e4s_21.02	e4s	Test umpire@4.1.2 E4S 21.02 test via spack test
spack_test_gasnet_e4s_21.05	e4s	Test gasnet@2021.3.0%intel with e4s/21.05 via spack test
spack_test_hypre_e4s_21.02	e4s	Test hypre@2.20.0 for e4s/21.02 test via spack test
spack_test_hypre_e4s_21.05	e4s	Test hypre@2.20.0 for e4s/21.05 test via spack test
spack_test_upcxx_e4s_21.05	e4s	Test upcxx@2021.03.0 for e4s/21.05 test via spack test
spack_test_raja_e4s_21.02	e4s	Test raja@0.13.0 e4s/21.02 test via spack test
spack_test_raja_e4s_21.05	e4s	Test raja@0.13.0 for e4s/21.05 test via spack test
spack_test_strumpack_e4s_21.02	e4s	Test strumpack@5.1.1 E4S 21.02 test via spack test
spack_test_strumpack_e4s_21.05	e4s	Test strumpack@5.1.1 for e4s/21.05 test via spack test
spack_test_qthreads_e4s_21.02	e4s	qthreads E4S 21.02 test via spack test
spack_test_mfem_e4s_21.02	e4s	Test mfem@4.2.0 for e4s/21.02 test via spack test
spack_test_mfem_e4s_21.05	e4s	Test mfem@4.2.0 for e4s/21.05 test via spack test
spack_test_superlu_e4s_21.02	e4s	Test superlu@5.2.1 E4S 21.02 test via spack test
spack_test_superlu_e4s_21.05	e4s	Test superlu@5.2.1 and superlu-dist@6.4.0 for e4s/21.05 test via spack test
moduletest_e4s_20_10	modules e4s	Run module load test for e4s 20.10 module tree

Closing Remarks

- The facility deployment of E4S impacts how tests are written. We need a spack instance for deployment in order to test the user-facing environment. At Cori we can load e4s via **module load e4s** which activate a spack environment pre-installed with e4s packages
- **spack test** and E4S Testsuite requires a spack instance to run tests which is focused on testing spack stack whereas buildtest is focused on writing facility tests
- Issues with **spack test** at Facility will be addressed in future spack release, though facility deployment will be fixed to version. In those case we need to develop tests at facility when appropriate or periodically rebuild software with new version
- Buildtest leverages *spack test* or *E4S Testsuite* to run the E4S tests targeting our e4s deployment at NERSC, when test fails we would write a facility flavored test.
- Tests may require need for batch submission for different schedulers and buildtest can support job submission for Slurm, LSF, PBS and Cobalt.
- Test needs to be run on recurrent basis and automation can be done through the use of Gitlab. Finally test results needs to be published somewhere to analyze results.
- There are different ways to pass test including: return code, regular expression, runtime. Every test would need some criteria for success or failure
- *We need a human to analyze test result and report issues for facility tests. We need help from Developers to help contribute test and analyze test results for facility results.*

References

- Buildtest Docs: https://buildtest.readthedocs.io/en/latest/index.html
- Schema Docs: https://buildtesters.github.io/buildtest/
- Installing buildtest: https://buildtest.readthedocs.io/en/latest/installing_buildtest.html
- Getting Started: https://buildtest.readthedocs.io/en/latest/getting_started.html
- References: https://buildtest.readthedocs.io/en/latest/references.html
- Slack: http://hpcbuildtest.slack.com/
- API: https://buildtest.readthedocs.io/en/latest/api/index.html
- Spack: https://spack.readthedocs.io/en/latest/
- E4S Testsuite: https://github.com/E4S-Project/testsuite

Acknowledgement

This research was supported by the Exascale Computing Project (17-SC-20-SC), a collaborative effort of two U.S. Department of Energy organizations (Office of Science and the National Nuclear Security Administration) responsible for the planning and preparation of a capable exascale ecosystem, including software, applications, hardware, advanced system engineering and early testbed platforms, in support of the nation's exascale computing imperative.

This research used resources of the National Energy Research Scientific Computing Center (NERSC), a U.S. Department of Energy Office of Science User Facility located at Lawrence Berkeley National Laboratory, operated under Contract No. DE-AC02-05CH11231.