Automated Software Testing of Spack/E4S with buildtest

Shahzeb Siddiqui (LBNL)

Venue: SC23
BoF: Software Testing for Scientific Computing in HPC
Nov 15th, 2023
Outline

• Deploying Software in HPC is complex
• Workflow for Automated Testing
• Buildtest
• Challenges we faced
• Key Takeaways
Deploying Software in HPC is complex

- Complex dependencies between software packages and the environment
- Incompatibility issues
- Lots of dependencies
- Testing is vital for providing high quality software
Workflow for Automated Testing
Buildtest

- **Buildtest** is a testing framework that builds and executes tests on your HPC systems.
- Intended for **HPC staff, developers, and end-users**.
- Tests are written in **YAML**.
- Integration with Batch Scheduler, modules, spack, CrayPE.

[https://github.com/buildtesters/buildtest](https://github.com/buildtesters/buildtest)
Challenges we faced

• Bugs in tests, CI job or buildtest
• High CPU utilization and long runtimes (~45min)
• Gitlab runner availability
• Non Deterministic output due to concurrency limit in CI jobs
• Failures in spack test can’t be fixed in production deployment
• User Environment (shell, startup configuration, modules) can impact test behavior
Key Takeaways

• Use a HPC testing framework (reFrame, buildtest, pavilion2) to conduct site-local testing

• Infrastructure issues are bound to happen. Communication amongst groups is essential

• Spack for automated testing has its issues
  – Single Repository: Not feasible for facilities to fix test issues that are site-specific
  – Not compatible with batch schedulers for testing on compute nodes
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.
Reference Slides

Slides: https://docs.google.com/presentation/d/15xW4tA2eoMatMZs1cqX2fxhbuQB2bWdv/edit?usp=sharing&ouid=10212664227037583807&rtf=0
Buildtest-NERSC: Post-Deployment Testing and Automation

Overview

- **Buildtest-NERSC** is a collection of tests (mostly Spack/E4S) that are performed after a software deployment to ensure its quality. It utilizes **buildtest** to manage and run these tests smoothly.

Operational Mechanism

- Utilizes Gitlab CI for automated test execution
- Scheduled cron runs for consistent testing
- Test results shared via CDASH
- Gitlab runners on Perlmutter and Muller to enable CI workloads
- All test are run using the default branch (devel)

[https://github.com/buildtesters/buildtest-nersc](https://github.com/buildtesters/buildtest-nersc)
Spack is a package manager for supercomputers.

Used to deploy our software stack.

Spack supports stand-alone testing which buildtest can leverage.
E4S Testsuite includes over 100 tests, with a goal of full test coverage for the E4S ecosystem.

- Test code largely extracted from package sources.
- Uses a simple, extensible bash-script test driver.
What does a typical build look like
CDASH - Web Based Dashboard

5 passed, 3 failed, 0 not run, 3 missing.

- CDASH produces color coded charts, timelines (track progress over time), and links to test output
- Comprehensive presentation of test results posted to the web for users and developers
- Buildtest can upload test results to CDASH

https://my.cdash.org/index.php?project=buildtest-nersc
Test Results and metadata captured in CDASH

Test: spack_test_gasnet_e4s_22.05 (Passed)
Build: e4s (permuttet) on 2023-08-03 19:20:22
Labels: e4s

Test content:

```
#!/bin/bash

<<<<<<< START OF PRE COMMANDS >>>>>>
module load e4s/22.05
spack env activate -v gcc
spack load --first gasnet

END OF PRE COMMANDS >>>>>>

source /global/common/software/spackrc/gasnet-e4s/22.05/7853/spack/share/spack/setup-env.sh
spack test run -alias 67983f6f-99e5-4d59-a5bf-c8c2f43bca77 gasnet
spack test results -l --gasnet
```

Test output:
```
```

View GitLab CI results

15