Exceptional service in the national interest

TESTING TRILINOS WITH SPACK

Samuel E. Browne and Justin M. LaPre

Trilinos User-Developer Group Meeting 11/02/2023

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology and Engineering Solutions of Sandia LLC, a wholly owned subsidiary of Honeywell International Inc. for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.

SAND2023-11771PE

Background on Trilinos Framework

Our team manages code that supports build/test/reporting frameworks for HPC scientific software applications. Of primary importance is management of built/tested configurations with respect to CI-style automated testing that protects the Trilinos project from changes which introduce regressions. In doing so, we have to manage a large level of complexity with respect to third-party dependencies, platform-specific requirements, desired features to test across multiple compiler/MPI toolchains, and the ability for developers to replicate results in their local environments.

Managing Configurations of Trilinos

- Currently use a home-grown system called ‘GenConfig’
- Use files populated with CMake variables that get abstracted into higher-level concepts

```
[rhel7_sems-gnu-8.3.0-openmpi-1.10.1-serial_debug_shared_no-kokkos-arch_no-asan_no-complex_no-fpic_mpi_no-pt_no-rdc_no-uvm_deprecated-on_all]
use rhel7_sems-gnu-8.3.0-openmpi-1.10.1-serial_debug_shared_no-kokkos-arch_no-asan_no-complex_no-fpic_mpi_no-pt_no-rdc_no-uvm_deprecated-on_no-package-enables
use PACKAGE-ENABLES|ALL
```

- Issues
  - Not easily accessible outside of Sandia
    - Several configuration repositories are internal-only
    - Depends on recognizing hostnames to identify valid environments
    - Tightly coupled with on-system modules for TPLs/compilers/MPIs provided by another team
  - Must add a complete new ‘spec’ to reflect a change in value of one ‘variant’ (e.g. turn on AddressSanitizer)
  - To be fair, most of these issues are resolvable

Enter Spack

Spack is a widely-adopted package manager that enables a uniform configuration interface for many HPC codes. Trilinos was chosen as an example codebase to showcase Spack's abilities. Instead of dealing with GenConfig and INI files, we can simply do this:

```
spack install trilinos
```

- Spack supports multiple versions of major compilers and MPI interfaces.
- Also supports "variants" e.g., `shared`, `hdf5`, `tempus`, etc.
- Adding variants is straightforward in the Python file:

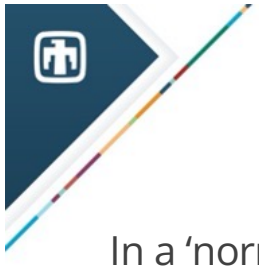
```
variant("openmp", default=False, description="Enable OpenMP")
```

Managing Configurations of Trilinos with Spack

```
[rhel7_sems-gnu-8.3.0-openmpi-1.10.1-serial_debug_shared_no-kokkos-arch_no-asan_no-complex_no-fpic_mpi_no-pt_no-rdc_no-uvm_deprecated-on_all]
use rhel7_sems-gnu-8.3.0-openmpi-1.10.1-serial_debug_shared_no-kokkos-arch_no-asan_no-complex_no-fpic_mpi_no-pt_no-rdc_no-uvm_deprecated-on_no-package-enables
use PACKAGE-ENABLES|ALL
```

```
trilinos %gcc@8.3.0 +debug+shared~complex~fpic+mpi~cuda_rdc~uvm ^openmpi@1.10.1
trilinos %gcc@8.3.0 +debug+shared~complex+mpi ^openmpi@1.10.1
```

- Issues
  - Not all required features current supported in package.py (e.g. building with AddressSanitizer)
  - Spack can feel like it adds overhead
  - Deployment is great, but how does one develop (including running tests) with Spack?



Testing Trilinos with Spack

In a 'normal' build of Trilinos, there is one more variant that is enabled: tests. Without a directive to build tests, Trilinos' build system does not build or enable testing (as with many CMake projects).

How would one do this with Spack?

```
spack install --keep-stage trilinos+test
spack stage trilinos+test
cd $(spack location --build-dir trilinos+test)
ctest …
```

Who would want to do this with Spack?

- Users installing Trilinos on systems who want to run the Trilinos tests for configuration validation (e.g. an external HPC system administrator)

Testing Trilinos with Spack – Developer Style

How could I do this same process with my own local Trilinos checkout?

```
$> git clone https://github.com/trilinos/Trilinos.git
$> cd Trilinos
$> # do a bunch of work
$> spack dev-build trilinos@develop+test ...
$> ctest ...
```

Who would want to do this with Spack?

- Developers who want to replicate Trilinos test failures who are already familiar with Spack

Mostly

^Live Demo