



Trilinos Framework Product Overview and Update



James Willenbring

Trilinos Framework Product Overview and Update



- Contributing
- Testing
 - Pull Request and Develop to Master
 - Other
- Releases
- Spack
 - E4S
 - xSDK
- Current and upcoming efforts

Contributing to Trilinos



<https://github.com/trilinos/Trilinos/blob/master/CONTRIBUTING.md>

- Create an issue
- Implement feature/fix/etc
- Create a Pull Request (PR)
 - PR will need to be reviewed and all PR testing must pass

The screenshot shows a GitHub pull request review interface. At the top, there's a red circle with a white 'X' labeled 'Review required' and a note: 'At least 1 approving review is required by reviewers with write access. [Learn more](#)'. To the right is a 'Show all reviewers' link. Below this, there's a section for '3 pending reviewers'. The next section, 'Some checks haven't completed yet', shows '3 expected checks' with three items: 'Pre-Merge Inspection' (status: 'Expected — Waiting for status to be reported'), 'Pre-Test Inspection' (status: 'Expected — Waiting for status to be reported'), and 'Pull Request AutoTester' (status: 'Expected — Waiting for status to be reported'). Each of these has a 'Required' button to its right. At the bottom, there's a red circle with a white 'X' labeled 'Merging is blocked' and a note: 'Merging can be performed automatically with 1 approving review'. At the very bottom, there are buttons for 'Merge pull request' and 'You can also open this in GitHub Desktop or view command line instructions.'



- Pull Request testing
 - A set of testing configurations that must pass before changes are made to the develop branch
 - PR test configurations
 - CUDA 10.2.2
 - CUDA 10.2.2 (UVM off)
 - gcc-8.3.0
 - gcc-7.2.0-serial
 - gcc-7.2.0-debug
 - Intel-17.0.1
 - Clang-10.0.0
 - Python-3



- Develop to Master promotional testing
 - A set of testing configurations that must pass before changes are promoted from the develop to the master branch
 - Run nightly. Includes all PR configurations (except Python-3) plus:
 - gcc-7.2.0
 - Intel-19.0.5
 - Clang-7.0.1
 - Clang-9.0.0
 - CUDA 10.1.243
 - CUDA 10.1.243 rdc



- Specific customer-focused testing
 - Not tied to branch promotions
 - Semi-manually triaged
 - A couple dozen builds are monitored by the framework team
- Other testing
 - Experimental
 - Package-owned
 - Customer-owned

<https://testing.sandia.gov/cdash/index.php?project=Trilinos>



- Renewing a focus on tagged releases
- Target going forward is to release quarterly
- One major release per year
- Major releases allow breakages in backward compatibility
 - No current tests for backward compatibility
- Releases branches are created from the master branch
 - If a release will support a specific customer, acceptance testing may be done prior to release
- Patch updates may be made to release branches
- Most recent release was Trilinos 13.2.0 in October

Building Trilinos Through Spack

- Sameer Shende will speak about E4S and Trilinos Thursday!
- E4S installation instructions
 - <https://e4s-project.github.io/manual-installation.html>
- E4S 21.11 environments
 - <https://github.com/E4S-Project/e4s/tree/master/environments/21.11>
- xSDK installation instructions
 - <https://xsdk.info/installing-the-software/>
- xSDK 0.7 platform files
 - <https://github.com/xsdp-project/installxSDK/tree/r-0.7.0/platformFiles>
- These instructions are general and do not support highly customized Trilinos builds

Current and Upcoming Trilinos Framework Efforts

- Finish moving to new GenConfig build infrastructure (see talk on Thursday)
- Ramp up on Spack and take ownership of the Trilinos Spack package
- Finalize move of PR CUDA builds to a new system and assess apparent capacity problems
- Minor/major release preparation
- Numerous build additions and tweaks
 - Caraway (in progress)
 - C++17 (started, but not currently being worked)
 - Intel 2021
 - CUDA 11
 - Memory leak testing
 - ...