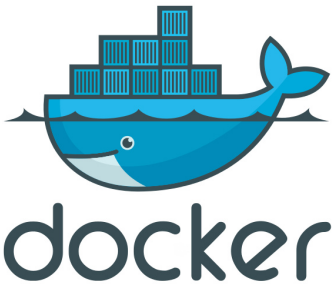
Exceptional service in the national interest

Sandia National Laboratories



Using Trilinos from Containers & Interactive Environments

Tobias Wiesner

May 12, 2019

U.S. DEPARTMENT OF ENERGY | National Nuclear Security Administration

Sandia National Laboratories is a multi-mission 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 a National Nuclear Security Administration under contract DE-NA0003525.
SAND NO. 2016-10805 C

Doing science in HPC

Sandia National Laboratories

Typical workflow from the first idea to the final performance tests:

1) Idea & Algorithm design

- Rapid development (Matlab, Python, C++)
- Quick experiments on small data sets

2) Implementation

- Final parallelized implementation in C++ (using Trilinos)
- Algorithmic optimizations & Performance optimizations

3) Performance studies on HPC clusters

- Final performance studies on HPC clusters with large datasets
- Produce results for scientific paper

Doing science in HPC

Sandia National Laboratories

What are the challenges?

1) Idea & Algorithm design

- Focus on algorithms and mathematics
- Need for rapid development tools and programming languages

2) Implementation

- Error-prone reimplementation and parallelization of algorithms using Trilinos
- Expert knowledge for best results necessary (Trilinos, etc.)

3) Performance studies on HPC clusters

- Make your algorithms work on all HPC platforms

Sandia National Laboratories

New technologies and tools

What is docker?

Sandia National Laboratories

Docker is a software containerization platform.

- New concepts for *organizing*, *developing*, *testing* and *shipping* software
- Maintain one general software/development environment independent of underlying host OS or hardware
- **Focus on the applications!**

More information: `https://docs.docker.com/install/`

What is jupyter?

Sandia National Laboratories

Jupyter is an interactive computing environment.

- Support for many language kernels including Python, Julia, R, C++
- Ideal tool for interactive tutorials
- Jupyter notebooks for documenting and organizing scientific experiments

Jupyter usage scenarios

Sandia National Laboratories

- Document your ideas, algorithms, experiments and results in notebooks
- Use functionality from existing C++ libraries in your notebooks
- Extend the functionality by writing new classes and functions live in your notebook
- Gain better insight into legacy code and complex algorithms by adding advanced debugging statements
- Optimize your algorithms in an interactive way

Example

`AutoWhiteBalancingLiveDemo.html`

What jupyter is and what it is NOT

Sandia National Laboratories

Jupyter

- helps you minimizing the effort for porting the Jupyter C++ implementation to the final implementation.
- helps you developing algorithms and gaining in-depth insight.
- helps you organizing your ideas, experiments and results.

Jupyter does not

- make porting the experimental C++ code to final C++ code superfluous.
- help you writing clean code and unit tests.
- help you organizing your notebooks. You still need a systematic way of storing your notebooks in order to quickly find results from a specific experiment. Use a version controlling system!

Sandia National Laboratories

New technologies in context of Trilinos

Doing science in HPC

Sandia National Laboratories

How can docker and jupyter help?

1) Idea & Algorithm design

- Use C++ with Cling and Jupyter as rapid prototyping environment?

2) Implementation

- Improve your Trilinos knowledge with Jupyter Trilinos tutorials
- Manage your development environment with docker

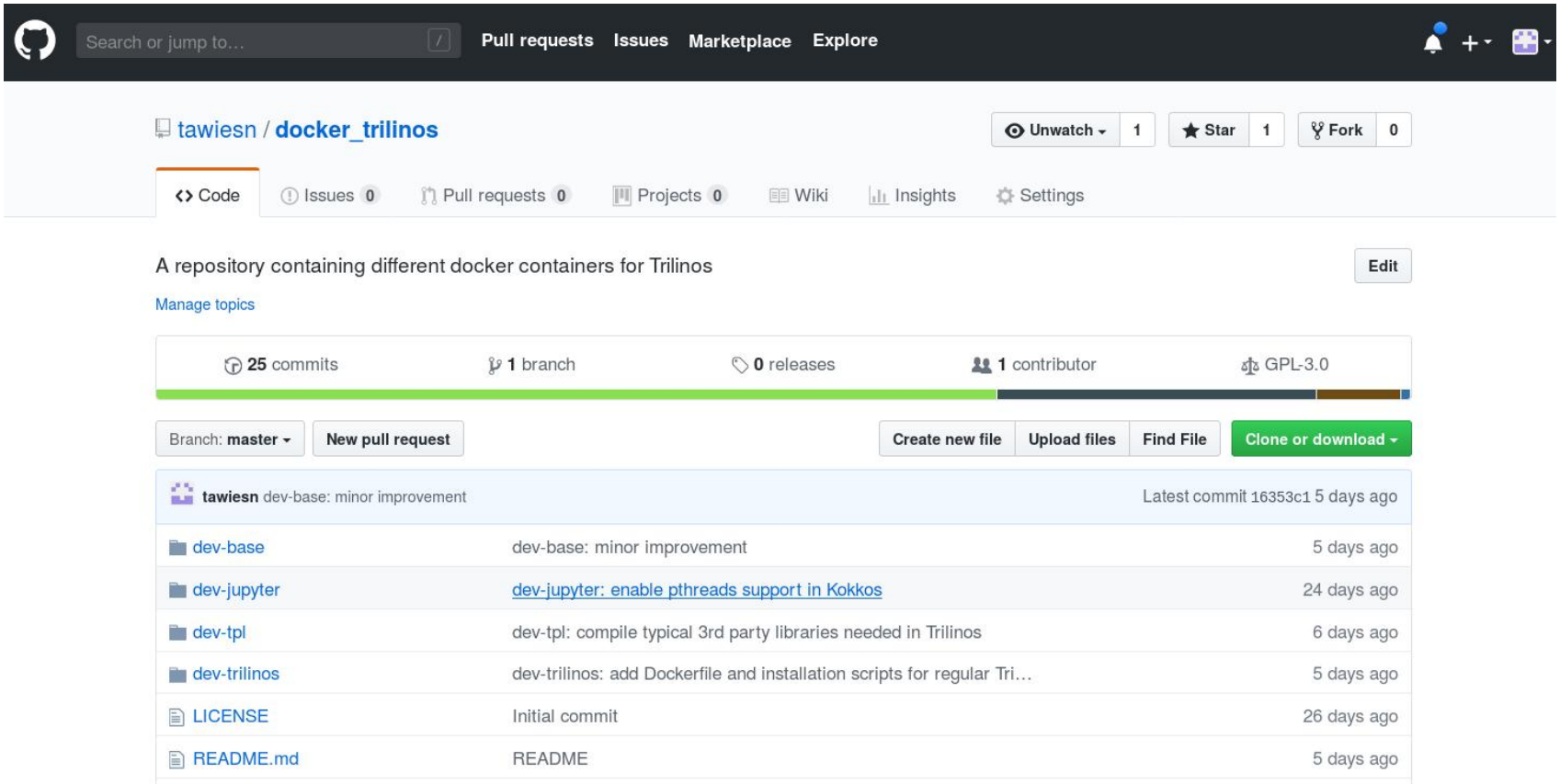
3) Performance studies on HPC clusters

- Manage your development environment with docker

github repository

Check out the github repository

`https://github.com/tawiesn/docker_trilinos`



Sandia National Laboratories

Docker for Trilinos

Trilinos docker container

Sandia National Laboratories

The dev-trilinos docker container

- Classical docker container based on CentOS 7 (Redhat compatible)
- Contains a clone of the public Trilinos repository
- Builds some core Trilinos packages using gcc with OpenMPI support
- **Meant as an example for your own Trilinos container** that can be used to link your application against
- Use it as **standardized development platform** that can also be used on **HPC clusters**, e.g. via the Shifter project (`https://github.com/NERSC/shifter`)

More information here: `https://github.com/tawiesn/docker_trilinos/blob/master/README.md`

How to create your own Trilinos container for your application

Sandia National Laboratories

- Clone the `https://github.com/tawiesn/docker_trilinos` repository
- Follow the instructions to build all dependencies. For details see `https://github.com/tawiesn/docker_trilinos/blob/master/installation_dev_trilinos.md`
- Create a copy of the dev-trilinos folder and rename it, e.g. my-dev-trilinos
- Edit the my-dev-trilinos/do-configure file and enable all Trilinos features that you need
- If you need a specific Trilinos version for your application you might adapt the git clone statement in the my-dev-trilinos/Dockerfile

How to create your own application docker container

Sandia National Laboratories

- Build your my-dev-trilinos docker container
- Create a new folder, e.g. my-application
- Create a new Dockerfile in you my-application folder
- Edit the my-application/Dockerfile and make sure you derive your docker container from the my-dev-trilinos folder.
- Copy the sources and build scripts into the docker container. Use the dev-trilinos/Dockerfile for some inspiration.

Benefit

Your Dockerfiles contain all necessary commands to compile and install your Trilinos libraries and application.

- Put your Dockerfiles under version control
- Set up nightly builds for the Docker container to track changes and detect compilation issues as soon as they pop up

Sandia National Laboratories

Jupyter and Trilinos

Trilinos jupyter notebooks

Sandia National Laboratories

The dev-jupyter docker container

- Docker container based on CentOS 7 (Redhat compatible) with clang installation and cling extensions
- Contains a clone of the public Trilinos repository
- **Meant for tutorials and algorithmic experiments**
- Find a small set of basic Trilinos tutorial notebooks here: `https://github.com/tawiesn/trilinos-notebooks`
- Limitations:
  - No OpenMPI support at the moment (can be added)
  - Some instabilities when running parallel Kokkos kernels (needs to be investigated)

More information here: `https://github.com/tawiesn/docker_trilinos/blob/master/README.md`

Basic Trilinos Jupyter mini tutorials

Sandia National Laboratories

Repository

`https://github.com/tawiesn/trilinos-notebooks`

1) Epetra tutorial

Apply power method to tridiagonal matrix

2) Belos tutorial

Solve linear equation with GMRES w/wo ILUt

3) Tpetra tutorial

Create Tpetra vector and access data

4) Kokkos tutorial

Some basic examples for parallel kernels and Kokkos views

Questions?

Sandia National Laboratories

For more information check out the repositories:

Docker container

`https://github.com/tawiesn/docker_trilinos`

Jupyter notebooks

`https://github.com/tawiesn/trilinos-notebooks`