

SANDIA REPORT

SAND2016-6771
Unlimited Release
Printed July 2016

Exploring Container Technologies for Large Scientific Libraries: Docker and Trilinos

Michael A. Heroux
James M. Willenbring
Sean J. Deal

Prepared by
Sandia National Laboratories
Albuquerque, New Mexico 87185 and Livermore, California 94550

Sandia National Laboratories is a multi-program laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

Approved for public release; further dissemination unlimited.



Sandia National Laboratories

Issued by Sandia National Laboratories, operated for the United States Department of Energy by Sandia Corporation.

NOTICE: This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government, nor any agency thereof, nor any of their employees, nor any of their contractors, subcontractors, or their employees, make any warranty, express or implied, or assume any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represent that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government, any agency thereof, or any of their contractors or subcontractors. The views and opinions expressed herein do not necessarily state or reflect those of the United States Government, any agency thereof, or any of their contractors.

Printed in the United States of America. This report has been reproduced directly from the best available copy.

Available to DOE and DOE contractors from

U.S. Department of Energy
Office of Scientific and Technical Information
P.O. Box 62
Oak Ridge, TN 37831

Telephone: (865) 576-8401
Facsimile: (865) 576-5728
E-Mail: reports@osti.gov
Online ordering: <http://www.osti.gov/scitech>

Available to the public from

U.S. Department of Commerce
National Technical Information Service
5301 Shawnee Rd
Alexandria, VA 22312

Telephone: (800) 553-6847
Facsimile: (703) 605-6900
E-Mail: orders@ntis.gov
Online order: <http://www.ntis.gov/search>



SAND2016-6771
Unlimited Release
Printed July 2016

Exploring Container Technologies for Large Scientific Libraries: Docker and Trilinos

Michael A. Heroux, James M. Willenbring
Center for Computing Research
Sandia National Laboratories
P.O. Box 5800
Albuquerque, New Mexico 87185-MS1320
Sean J. Deal
Department of Computer Science
Saint John's University
Collegeville, MN 56321

Abstract

The Trilinos Project [1] produces, maintains and distributes a large collection of reusable, parallel scientific libraries. Docker [2] provides container technologies that support compilation, packaging, distribution and execution of software on Linux, Mac OS and Windows systems¹, with emerging support for Cray platforms [3]. In this short article we describe recent efforts to explore the potential for using Docker in a variety of settings to enhance several Trilinos Project workflows. The technical foundation for this article is presented in an Honors thesis of one of the authors [4].

¹ Linux support is native. Native MacOS and Windows support is available in pre-release form.

ACKNOWLEDGMENTS

The authors thank the ASC program that funded this work.

CONTENTS

1. Introduction.....	7
2. Docker Possibilities	9
3. Trilinos Docker Effort Status.....	13
3. Summary	15
4. References.....	17
Distribution	18

FIGURES

Figure 1: These timing results compare the execution measured in MFLOP/s (up is good) of key performance-impacting kernels using a native installation (blue bars) to a Docker installation (orange bars). These results are from a collection of 48-process MPI runs on the 8 node Melchior cluster at St. John’s University. The work was distributed with six MPI ranks per node. Timings show that executing in Docker containers and using message passing across containers did not qualitatively impact performance. For reasons we do not understand, we even see better performance from the Docker version in some cases. This table, and further experimental results, are available the Deal thesis [4]..... 10

NOMENCLATURE

DOE	Department of Energy
Image	A binary file containing a collection of software
Container	A running instance of an image that can be used for computing

1. INTRODUCTION

The Trilinos project produces and supports a large and growing collection of reusable software components called *packages*. Package development teams are semi-autonomous, and some Trilinos-compatible packages are developed completely independently. But the majority of packages (approximately 60) are part of a single Git repository hosted on GitHub.com². The core Trilinos user base is composed of application developers running on parallel distributed memory systems with multicore, manycore and GPU devices. This user base is accustomed to forking from the Trilinos repository or downloading one of our compressed source tarfiles (released approximately quarterly), and then building the libraries, tests and examples on their native system.

While this process, which is very common in the computational science and engineering (CSE) community, works for many users and provides a lot of flexibility in deciding which Trilinos package are built, which compilers, third-party libraries and optimization flags are used and more, it is daunting, especially for new users. Trilinos source code is primarily C++ and some packages use advanced features of the language, so it is not uncommon to encounter build errors with often cryptic error messages. Even when builds complete, compilation times can be long when using some of the template-based packages.

² <https://github.com/trilinos/Trilinos>

2. DOCKER POSSIBILITIES

The Trilinos Project has used virtual machine environments such as VirtualBox and VMWare for many years, but there have always been technical (performance) or access (license) barriers to promoting these virtual environments. Docker promises to be different by preserving performance, both in execution time and storage size, and being widely available without restrictions.

Trilinos Tutorial via Docker. We explored Docker first as a way to package and distribute the web-based Trilinos tutorial (WebTrilinos). Prior to using Docker, using the tutorial required installation on a networked workstation with password controlled access. Docker is attractive because installing the tutorial was very challenging, requiring careful configuration of the web portal, special security privileges, and access monitoring. Using the Docker container, we can now have individual installations of WebTrilinos through a simple two-step process: Install Docker, then install the WebTrilinos container³. Once installed, the user can launch tutorial webpage in their native web browser, or step through exercises with the command line interface.

Reference Development and User Environment. We are presently exploring the use of Docker to provide a portable reference Trilinos development and user environment. While the Trilinos project tries to support as many computing platforms, operating systems and compiler environments as possible, there is tremendous value in having a universally accessible reference environment. Such an environment permits better collaboration on detecting and correcting software errors and can also be built to contain the many third party libraries that can be used via Trilinos, pre-built so that a developer can simply use them. This environment is also very useful as the foundation for a pre-check testing platform.

Standard Binary Distributions. One of the most obvious values of a pre-built version of the Trilinos libraries is its use by applications as the primary binary for linking. We anticipate that many of our users who currently rely upon a particular numbered Trilinos release (e.g., Trilinos 12.6.1), could instead rely upon a Docker container with a pre-built instance of the same version.

Specialized Containers. Trilinos can be used to solve many kinds of problems, but to understand the available capabilities requires a significant effort. One must learn about many packages, parameters and data structures in order to be able to construct a solution strategy for their specific problems. For example, if someone wants to compute a truncated singular value decomposition (SVD) of a large matrix, they must understand that Anasazi is the Trilinos eigensolver package, that it can be used to compute a truncated SVD, and then must form a distributed matrix, or provide a function for “matrix-free” application of the linear operator to a vector.

Using Docker and a modest driver program, we can provide a turn-key solver for large-scale SVD computations, and package it in a Docker container. The interface to the driver would be simple, requiring only the specific required for computing the SVD, and asking the user to provide matrix data in simple arrays or, as an advanced option, a function that would, when called by Trilinos, compute the action of the linear operator on a vector. These specialized containers can be valuable

³ <https://hub.docker.com/r/sjdeal/webtrilinos>

to our traditional CSE user community, but should be of particular value to data sciences communities who are, as we understand, particularly accustomed to accessing software capabilities in this way.

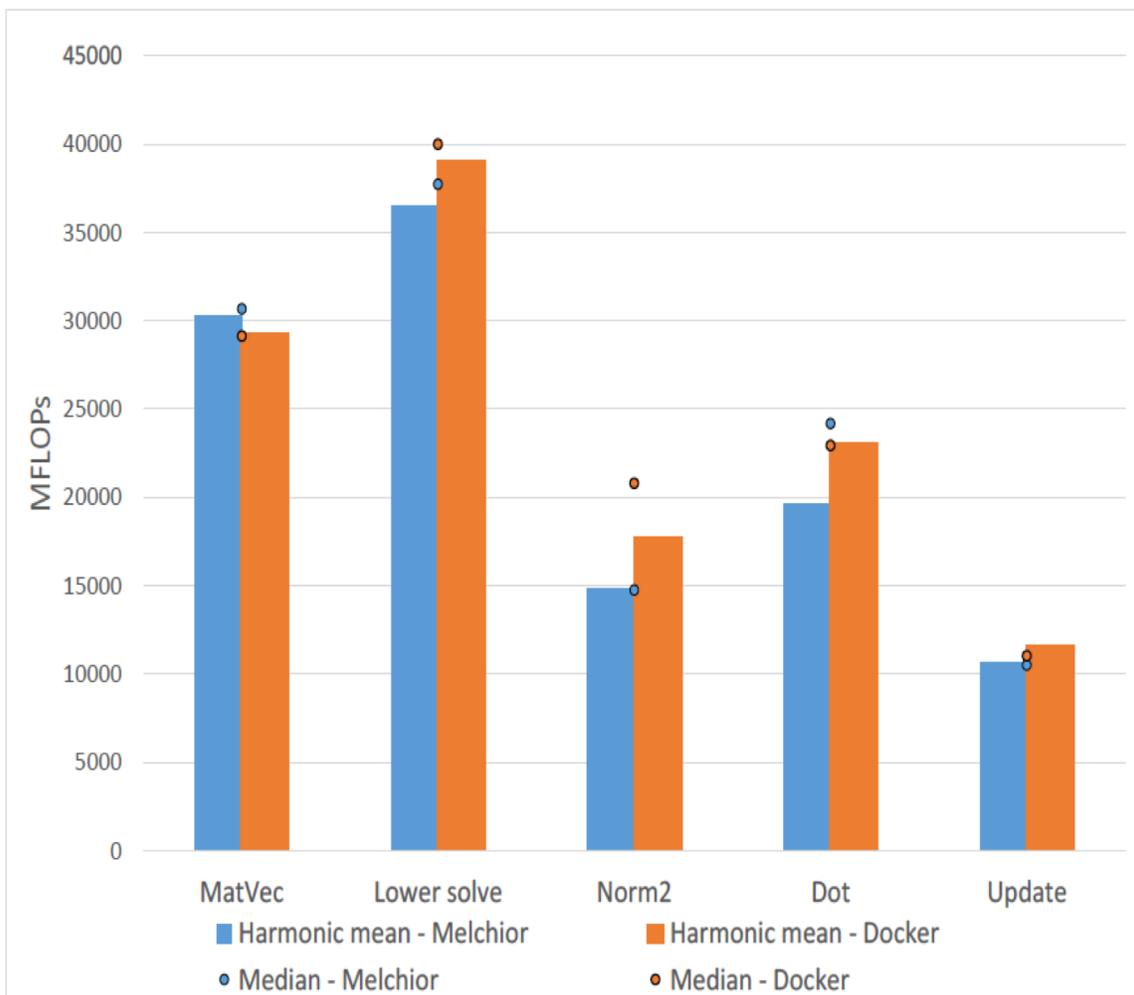


Figure 1: These timing results compare the execution measured in MFLOP/s (up is good) of key performance-impacting kernels using a native installation (blue bars) to a Docker installation (orange bars). These results are from a collection of 48-process MPI runs on the 8 node Melchior cluster at St. John’s University. The work was distributed with six MPI ranks per node. Timings show that executing in Docker containers and using message passing across containers did not qualitatively impact performance. For reasons we do not understand, we even see better performance from the Docker version in some cases. This table, and further experimental results, are available the Deal thesis [4]

Portable High Performance. As demonstrated in Deal’s thesis [4] (see Figure 1) and with Cray’s intent to support Docker, there is a realistic possibility that Docker containers can provide performance portability on many platforms, even those with GPUs and other advanced computing devices. The value of this potential cannot be overstated, and could truly revolutionize the packaging and distribution of scientific and engineering software, enabling qualitatively better

computing workflows. While realizing the full potential of this approach is still some time away, there has been tremendous progress, even within the past year. The Trilinos project will continue to pursue this capability aggressively.

New Scientific Workflows. The ability to run a Docker container from the host, via command line, script or batch mode, provides an easy workflow management environment. Specialized Docker containers that provide a simple interface to key functionality in a multi-purpose library. Using a collection of special purpose containers such as the SVD solver mentioned above, users can create sequences of operations to support a sophisticated workflow for solving a scientific problem using Docker containers and simple scripts.

3. TRILINOS DOCKER EFFORT STATUS

The Trilinos project has already seen benefits from using Docker for its WebTrilinos distribution, and we have measured excellent performance on a Linux cluster. We will continue these explorations.

Our present challenges with Docker include:

1. **Native Windows and MacOS support.** Native Windows and MacOS support is still in the beta testing phase. While we have had some success on MacOS (we have not tried the native Windows version), installation is still not seamless.
2. **Using Docker with a VPN.** Early experiences trying to use Docker while connected to a virtual private network (VPN) have not been successful.
3. **Docker container composition.** We can vertically stack Docker containers. But it would be useful if there was more support for generally combining containers. Not in a service sense, but in the sense of being able to deliver several software packages via one container. This can probably be scripted, but the ability to do this flexibly would be useful. There may be a solution to this issue of which we are unaware.
4. **Docker stability, maturity and support.** While we have a lot of excitement about Docker, we have found it to be surprisingly unstable. We have experienced the website being completely unavailable, and have had updated (non-beta) versions of Docker not work at all. We have had trouble with the beta version too, but expected that. It is also surprisingly difficult to find information when encountering problems. Basic info is well organized, but we had anticipated more discussion help threads directly applicable to problems we have had.

3. SUMMARY

The Trilinos Project has experimented with Docker as a packaging and distribution technology in several settings. Despite some instabilities, the emerging features and community support are very promising. If Docker fulfills its full potential, we believe its use in CSE can qualitative improve scientific productivity. Furthermore, its use can provide a much-needed bridge between the CSE and data sciences communities, providing access to the scalable computational capabilities that have long been a part of the CSE software ecosystem to the data sciences.

4. REFERENCES

- [1] Trilinos Project, "The Trilinos Project Main Page," 18 May 2016. [Online]. Available: <https://trilinos.org>.
- [2] Docker, Inc., "Docker - Build, Ship, and Run Any App, Anywhere," Docker, 18 May 2016. [Online]. Available: <https://www.docker.com>. [Accessed 18 May 2016].
- [3] Cray, Inc, "Cray - Investors - News Release," Cray, Inc., 17 November 2015. [Online]. Available: <http://investors.cray.com/phoenix.zhtml?c=98390&p=irol-newsArticle&ID=2112970>. [Accessed 18 May 2016].
- [4] S. J. Deal, "HPC Made Easy: Using Docker to Distribute and Test Trilinos," DigitalCommons@CSBSJU, Collegeville, 2016.

DISTRIBUTION

1 MS0899 Technical Library 9536 (electronic copy)



Sandia National Laboratories