SAND2018-2354R

EXASCALE COMPUTING PROJECT

ECP-U-2017-XXX

Kokkos User Support Infrastructure

WBS STPM12 Milestone 2

Authors

Christian Trott (1), Graham Lopez (2), Galen Shipman (3)

Author Affiliations

(1) Sandia National Laboratories

(2) Oak Ridge National Laboratory

(3) Los Alamos National Laboratory

February 28th 2018

U.S. DEPARTMENT OF ENERGY | Office of Science

National Nuclear Security Administration

EXASCALE COMPUTING PROJECT

EXECUTIVE SUMMARY

This report documents the completion of milestone STPM12-2 Kokkos User Support Infrastructure. The goal of this milestone was to develop and deploy an initial Kokkos support infrastructure, which facilitates communication and growth of the user community, adds a central place for user documentation and manages access to technical experts.

Multiple possible support infrastructure venues were considered and a solution was put into place by Q1 of FY 18 consisting of (1) a Wiki programming guide, (2) github issues and projects for development planning and bug tracking and (3) a “Slack” channel for low latency support communications with the Kokkos user community. Furthermore, the desirability of a cloud based training infrastructure was recognized and put in place in order to support training events.

U.S. DEPARTMENT OF ENERGY | Office of Science

NNSA National Nuclear Security Administration

1. INTRODUCTION

In order to develop a healthy community around the Kokkos programming model it is necessary to put a support infrastructure in place. That infrastructure must serve the users needs in finding information about Kokkos, get connected to experts to answer questions about how to use Kokkos as well as provide mechanisms to report bugs and request new capabilities. One key aspect of such an infrastructure is that it should be scalable, i.e. that the approach works in the early stages with just a few users as well as in later stages, where Kokkos may have thousands of active users.

2. MILESTONE OVERVIEW

2.1 DESCRIPTION

Develop and deploy Kokkos support infrastructure to facilitate:

- Communication and growth of the user community
- Repository for user documentation
- Managed access to technical experts.

2.2 EXECUTION PLAN

In this milestone, options for deploying a user communication strategy as well as documentation resources are explained, and selected. The infrastructure is put in place, which then can be filled with content by Kokkos developers as well as members of the larger Kokkos community. Communication tools are explored to facilitate low latency feedback mechanisms.

2.3 COMPLETION CRITERIA

There are three completion criteria:

(1) Infrastructure is in place to provide Kokkos documentation.

(2) Infrastructure is in place to allow users to report bugs, ask for new features, and find out about the status of said bugs and features.

(3) A mechanism is set up to facilitate direct user-to-user communication and user-to-developer communication.

3. TECHNICAL WORK SCOPE, APPROACH, RESULTS

A support infrastructure has a number of goals: (1) it should provide users with the information they need in a timely fashion, (2) it should be scalable with the growth of the user base, (3) and it must be maintainable by the developer team. In particular, (3) drove the final decision on how and where to

maintain resources. We in particular explored the option of hosting documentation on new websites, potentially with mechanisms to file requests. But Kokkos developers were already using github to track development efforts. Adding another site to monitor and interact with, would have negatively impacted their participation rate. Instead, the decision was made to integrate support efforts as much as possible in the existing workflows. For that reason, the majority of the support infrastructure is collocated with the Kokkos github repository. In addition, we introduced the utilization of a chat service called Slack for low latency communication. Again, Slack has the advantage that it is already in use by multiple ECP application teams, and thus by a number of Kokkos developers, increasing the participation rate of both Kokkos developers and users.

3.1 GITHUB WIKI

Github supports a number of mechanisms through which documentation and user communication can be facilitated and organized. For regular documentation each repository comes with wiki pages, which in fact live in their own repository. Those wiki pages are written using "markdown", a lightweight markup language which is readable as plain text as well as being automatically rendered to online content by the Github wiki infrastructure. We converted a previously existing programming guide written using LaTeX into markdown and updated it with the changes which happened in the 2 years since it was first generated. The programming guide gives an introduction into Kokkos with detailed explanations of the semantics of the programming model. We also put the infrastructure in place for a API reference for quick syntax lookups. The outline for the wiki pages is:

- Programming Guide
  - Core Programming Model
    - Introduction
    - Machine Model
    - Programming Model
    - Compiling
    - Initialization
    - View
    - Parallel Dispatch
    - Hierarchical Parallelism
    - Custom Reductions
    - Atomic Operations
    - Subviews
    - Interoperability
  - Containers
    - Dual View
    - Dynamic Rank View
    - Dynamic Length View
    - Unordered Map
  - Algorithms
    - Sorting
    - Random Numbers

- API Reference
  - Core
    - Initialization
    - View
    - Data Parallelism
    - Execution Policies
    - Memory Spaces
    - Task Parallelism

While the Programming Guide is more of a tutorial or textbook-like introduction to using Kokkos, the API reference is modelled more along the lines of widely used webpages such as cppreference.com or cplusplus.com. It is intended for users who are familiar with Kokkos but need to quickly look up syntax and restrictions of functions and classes provided by Kokkos. The team put the infrastructure for this API reference in place and generated reference documentation, with additional documentation continually being added and updated.

3.2 GITHUB ISSUE

"Github issues" are used for feature requests, bug reports, and lengthy general questions. These issues are typically triaged by a member of the development team and then assigned to the right developer to answer questions or address the reported bug. If actual development work has to be done, the issue is assigned to a milestone corresponding to the release cycle where a resolution is expected.

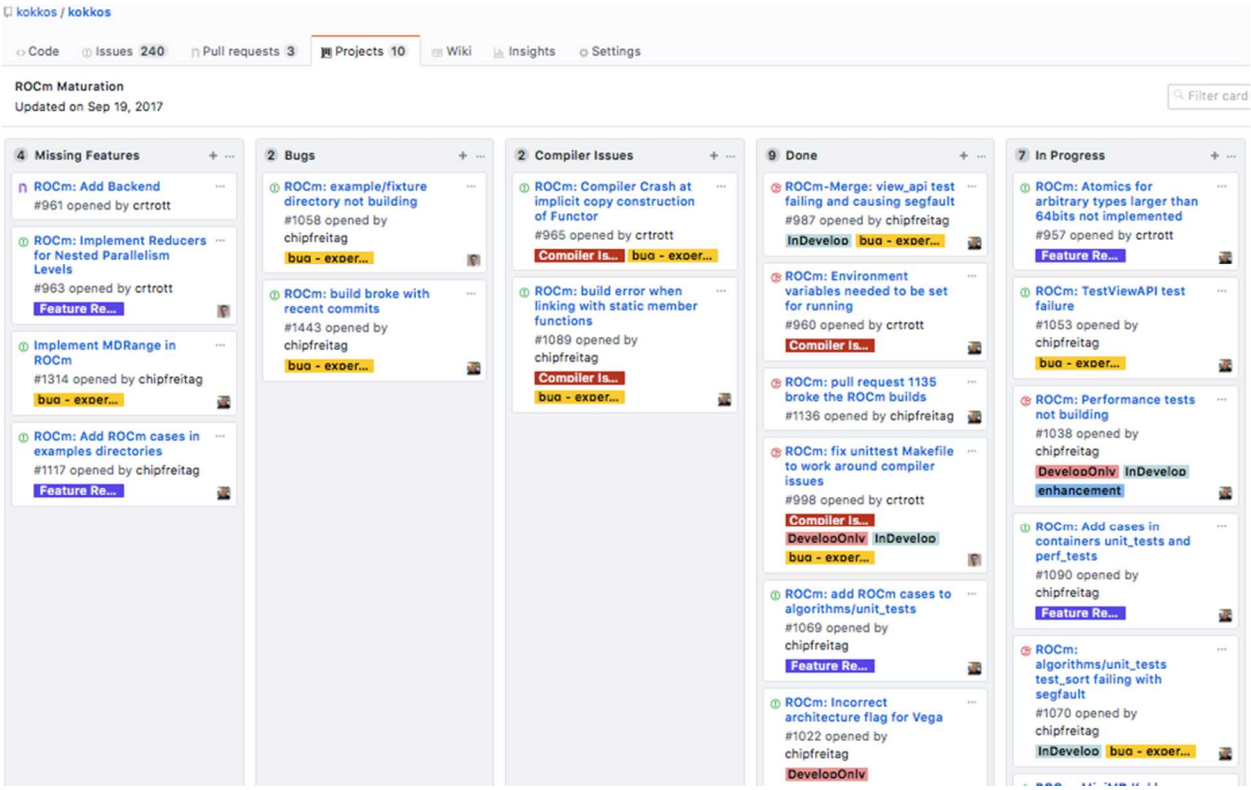
All issues are categorized based on their nature:

1) ***Feature Requests:*** These are big development efforts potentially taking multiple weeks of development time to finish. New capabilities fall into this category.
2) ***Enhancements:*** smaller development efforts taking a few hours to a few days of development time. Often, providing support for some previously not-supported corner case in the usage model of a Kokkos feature fall into this category.
3) ***Bug***: erroneous behavior of Kokkos which needs to get fixed.
4) ***Question:*** a user request for help or clarification. These are usually help request to get something working, or for implementation and optimization hints.
5) ***InDevelop:*** this label is applied to features which are now available in the semi stable "develop" branch of Kokkos. Features, Enhancements or Bugs marked this way are ready for testing by users.

Issues are "closed" as part of the release process of Kokkos during which the develop branch is merged into the master branch. A tool called github_changelog_generator (https://github.com/skywinder/github-changelog-generator) is then run to collect all closed issues and generate the Changelog for the Release. Typically it is necessary to manually prune the generated Changelog to filter out tightly related issues as well as internal tracking issues which are not relevant for users.

Large feature request or work items are broken down into smaller issues and all of them are assigned to a "github project". These projects are used to bundle related issues so that it is easier to track progress to the final goal. Depending on the complexity of the effort a Kanban-esque management process may be used for this subproject. This helps developers not only organize their work, but it also

allows users to gain insight into the development process and when certain features may be expected to be available. An example of such a project is the addition of a new backend to support AMD GPUs. In this case, issues are sorted into five columns: (i) Missing Features, (2) Bugs, (3) Compiler Issues, (4) In Progress and (5) Done.



3.3 SLACK CHAT CHANNEL

While these github mechanisms are great to track larger issues, they are somewhat cumbersome for rapid back and forth of questions and answers. To help with such communication scenarios, we created a “slack” channel. Slack is a chat tool with support for thematically organized channels, source code formatting and the ability to see the recent history for anyone with access to the channel. Slack works from within an internet browser but also provides desktop and smartphone apps, giving ample pathways to participate. In other communities, for example OpenACC, Slack has proven to be a very scalable tool with users eventually starting to answer questions of other users. One of our aims is to replicate that community development success, to free Kokkos developers from routine support questions so they can focus more on development work.

3.4 CLOUD INSTANCES

Another task which was tackled in the process of building up the community support infrastructure was the creation of Amazon Cloud instances with the Kokkos software stack being pre-installed. These instances provide a very straightforward way of starting to experiment with Kokkos without the need to make sure that compilers and drivers are up to date. It also facilitates access to GPUs for developers who might not otherwise have easy access. For now, the main use case of those instances is the deployment during training events. Instances are easily scalable, and dozens can be spun up within just a couple of minutes. Only a ssh-key and an IP address have to be sent to training participants in order to provide access. This approach incurs significantly lower administrative overhead than providing accounts on DOE

resources. Even if all class participants are DOE employees, for the training it is useful for everyone to have access to the same kind of machine so that all participants see the same compute environment. But getting compute accounts on another DOE facility can take multiple weeks even for DOE employees.

Going forward the cloud instance could be developed into a full fledged container solution, which would be available to every potential member of the Kokkos community. This would potentially lower the entry barrier to further experimentation with Kokkos locally, and thus allow for a more healthy growth of the community.

3.5 RESOURCE ACCESS

Most of the resources described are openly accessible. The only exception is the Amazon cloud instances, which are currently only used during training events organized by the Kokkos Support team.

Feature	Location	Access Restriction
Wiki Pages	https://github.com/kokkos/kokkos/wiki	Open access for anyone in the world.
Issues	https://github.com/kokkos/kokkos/issues	Read access for anyone in the world, creating and responding to issues requires a free github.com account.
Slack	https://kokkosteam.slack.com	Registration with slack is necessary. If registration is done with an email address from a national laboratory (lbl.gov, sandia.gov, lanl.gov, llnl.gov, anl.gov, ornl.gov) no approval is necessary to join the Kokkos slack channel. Other users must request access, which then is granted or denied by a Kokkos team member.

4. RESOURCE REQUIREMENTS

The work performed here required 0.25 FTE distributed over the three participating Labs.

5. CONCLUSIONS AND FUTURE WORK

The Kokkos Support project developed the infrastructure to facilitate communication between users, as well as between users and Kokkos developers. That infrastructure consists so far of wiki pages with programming guides and a space for API reference, github issues and github projects for organizing development plans, bug tracking and feature requests and a Slack chat channel to facilitate low latency communication. Additionally, the project developed container instances for the Amazon Cloud Service to facilitate training events. This infrastructure serves as the basis to develop and grow the Kokkos user community going forward. In particular it puts the mechanisms in place to grow the available documentation and provide a better knowledge base for current and future users of Kokkos. The container mechanism will potentially allow us to provide turn-key software development kits, in particular for

educational purposes. For example some universities are starting to use Kokkos in their class material, providing container solution will allow students to simply start developing Kokkos code on their existing machines without worrying about getting compatible compilers and other parts of the software environment.

6. ACKNOWLEDGMENTS

This research was supported by the Exascale Computing Project (ECP), Project Number: 17-SC-20-SC, a collaborative effort of two DOE organizations—the 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—to support the nation's exascale computing imperative.

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-NA-0003525.

This material is based upon work supported by Oak Ridge National Laboratory, which is operated by UT-Battelle, LLC., for the U.S. Department of Energy under Contract DE-AC05-00OR22750.

This work was performed under US Government contract DE-AC52-06NA25396 for Los Alamos National Laboratory, which is operated by Los Alamos National Security, LLC, for the U.S. Department of Energy.