SAND2019-13461R

EXASCALE COMPUTING PROJECT

Kokkos Training Bootcamp

WBS STPM12 Milestone 17

Authors

Christian Trott (1)

Author Affiliations

(1) Sandia National Laboratories

September 5th, 2019

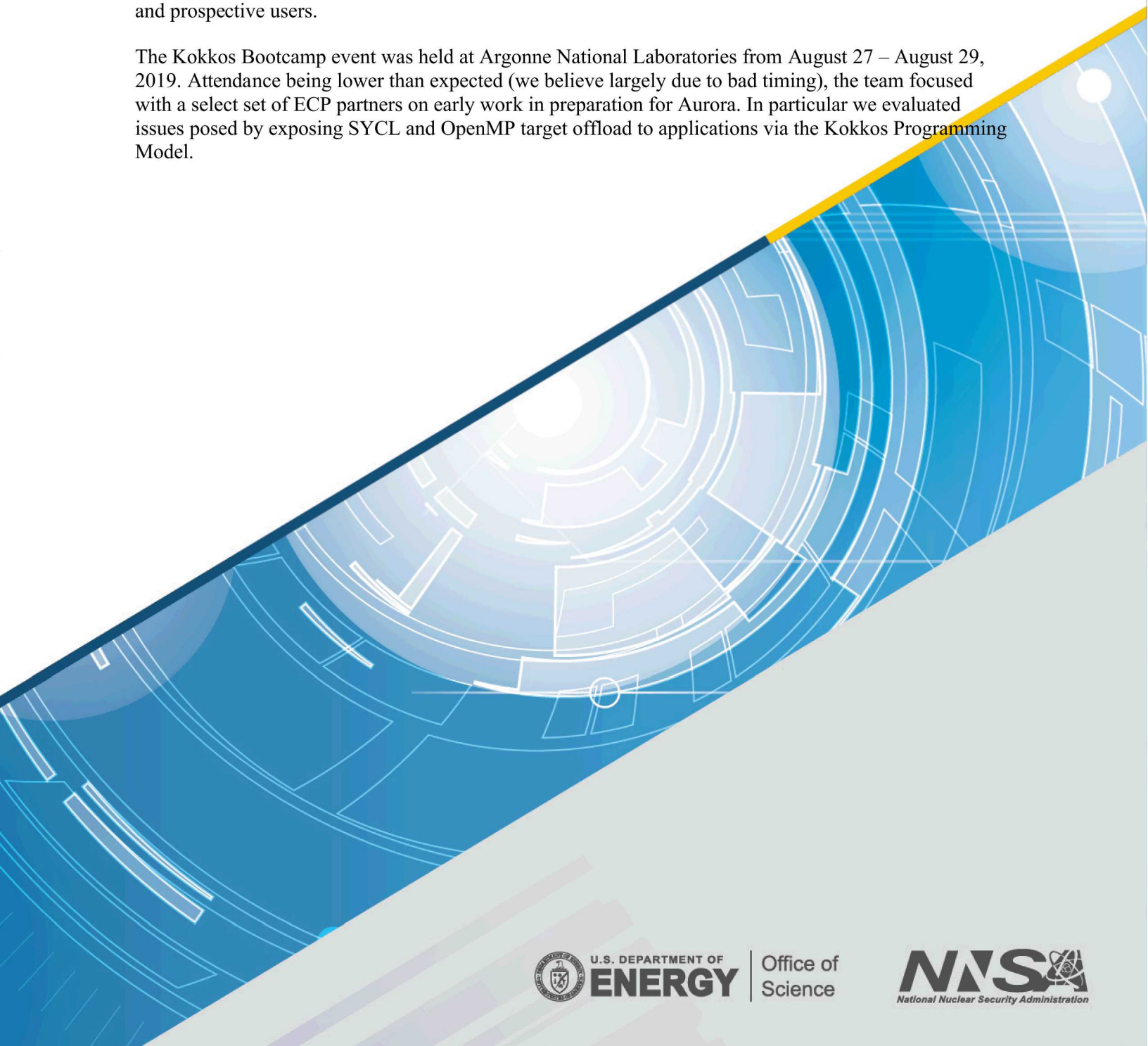
Sandia National Laboratories is a multimission laboratory managed and operated by National Technology & Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.

EXASCALE COMPUTING PROJECT

EXECUTIVE SUMMARY

This report documents the completion of milestone STPM12-17 Kokkos Training Bootcamp. The goal of this milestone was to hold a combined tutorial and hackathon bootcamp event for the Kokkos community and prospective users.

The Kokkos Bootcamp event was held at Argonne National Laboratories from August 27 – August 29, 2019. Attendance being lower than expected (we believe largely due to bad timing), the team focused with a select set of ECP partners on early work in preparation for Aurora. In particular we evaluated issues posed by exposing SYCL and OpenMP target offload to applications via the Kokkos Programming Model.



U.S. DEPARTMENT OF ENERGY | Office of Science

National Nuclear Security Administration

1. INTRODUCTION

Adopting a new programming model in HPC codes typically requires a very large development effort by the application experts and progresses in multiple stages from initial evaluation and experimentation to basic porting then to performance optimization. Further, the Kokkos user base consists of developers with a wide range of experience and expertise in programming with Kokkos, even sometimes within the same application development team. By providing Bootcamp training events that start from the basics of Kokkos use, but also allow for significant one-on-one time with Kokkos experts, developers with a wide range of experience can make progress during any stage of their adoption efforts.

2. MILESTONE OVERVIEW

2.1 DESCRIPTION

The goal of this milestone is organize and conduct Kokkos training bootcamp for ECP projects, which is open to other DOE applications on a space-available basis.

2.2 EXECUTION PLAN

In this milestone, we conducted a 3 day training event that including tutorials, hands-on exercises, and application-focused work on any code of participants' choosing.

2.3 COMPLETION CRITERIA

The completion criteria is to host the training event and report on the execution.

3. TECHNICAL WORK SCOPE, APPROACH, RESULTS

Kokkos support/outreach in ECP is centered around a number of related activities. Hackathons, tutorials, and other training events are a critical component of our outreach plan, affording us the opportunity to engage directly with current and potential users of Kokkos. To better facilitate ongoing engagement and responsiveness to our users we continue to enhance our support infrastructure. Finally, we solicit regular feedback from the ECP community both through our outreach/training activities and through direct application surveys.

In the runup to the tutorial it became clear that attendance would be comparatively low, which we suspect is partly the fault of scheduling mistakes. In particular the event was taking place just before the labor day weekend, where apparently numerous people planned to spend some extra vacation to make use of it. Adapting to these circumstances, we reduced the number of Kokkos team members attending and refucosed some of the effort on working on Aurora related activities – in particular since one of the projects attending (Balint Joo from the QCD community) is slated to be an early science code on Aurora.

3.1 SCHEDULE

The event was split between attendees who were new to Kokkos and experienced users.

	Track 1: Beginners	Track 2: Experienced
Day 1	Introduction Tutorial	Performance Analysis
Day 2	Intermediate Tutorial including Profiling	Performance Improvements / OpenMP backend discussions with compiler developers
Day 3	Hands-On work on customer apps	Aurora SYCL investigation

Kokkos Team Members and Contributers available: Christian Trott, Damien Lebrun-Grandie, Dan Sunderland, and Nevin Liber.

3.2 KOKKOS TUTORIAL

The tutorial portion of the event consisted of instructor-led lecture intermingled with hands-on exercises. Amazon AWS cloud instances with Volta V100 GPUs and the entire Kokkos software stack and tutorial materials pre-installed were used for the tutorials. This provided a very efficient method for getting ~40 participants started on actual coding and development as quickly as possible without having to troubleshoot login issues or installation and setup.

3.3 PERFORMANCE IMPROVEMENTS IN KOKKOS

Identified in an QCD miniapp (dslash), a problem with Kokkos::complex was diagnosed. On NVIDIA GPUs the compiler would issue 2 32bit load and store instructions instead of 1 64 bit instruction for a Kokkos::complex<float>. Replacing it with a custom complex type based on CUDA's float2 type increased performance by about 15%. Further investigation revealed that the main culprit is alignment assumption. Adding an alignment requirement in the Kokkos::complex type was able to fix the problem, which led to full recovery of performance. This change will be part of the Kokkos 3.0 release. These results were also used as part of a workshop paper for SC19. The paper was accepted and will be published as part of the Performance Portability and Productivity in HPC (P3HPC) workshop proceedings.

3.4 AURORA BACKEND INVESTIGATIONS

Two backends are proposed for Aurora: one based on SYCL and one on OpenMP target. We spend a significant amount of time on both working with experts local at Argonne and application team members. On the OpenMP side the most important discussions were with respect to the handling of nested parallelism. Currently the treatment in particular of #pragma omp simd is different in each compiler – some ignore it and some use it in a way that it is prohibitive expensive for our use cases. The main issue is how to use the SIMD construct. On the GPU Kokkos uses sub warp level parallelism for the "vector" level. This turned out to be very useful to extract parallelism from inner loops which may have only little work per iteration. But because the threads are all active from the entry into the kernel, certain statements such as atomics must be protected by a "single" construct outside of the inner loops – ensuring only one vector lane per threads executes the statement. OpenMP could mimic that behavior, but without the need

for the explicit single statement, since the compiler could mask out the problematic instructions. If that was the way the OpenMP offload compilers would behave the mapping of Kokkos to OpenMP would be straightforward as illustrated in the following table:

Kokkos	OpenMP

parallel_for("Loop", TeamPolicy<>(N,AUTO,8),
  KOKKOS_LAMBDA (const team_t& team) {
  int i = team.league_rank();
 	
  //Code1...
  parallel_for(TeamThreadRange(team,0,M),
  [&] (int j) {
  //Code2...
  parallel_for(ThreadVectorRange(team,K),
  [&] (int k) {
  //Code3...
  });
  });
});
	#pragma omp target teams distribute simd_length(8)
	for(int i=0; i<N; i++) {
	  #pragma omp parallel
	  {
	  //Code1...
	  #pragma omp for
	  for(int j=0; j<M; j++) {
	  //Code2...
	  #pragma omp simd
	  for(int k=0; k<K; k++) {
	  //Code3...
	  }
	  }
	  }
	}

Representatives of the ECP LLVM project agreed that this was a sensible interpretation of what to do with the SIMD construct and will explore implementing it.

Working on SYCL ended up mainly as a compiler debugging exercise. The availability of both a SYCL and a Kokkos version of DSLASH (a QCD miniApp) greatly furthered that effort, by enabling comparison of what should happen. We were able to narrow down a compiler issue which for now prevents progress with the Intel toolchain. Intel is working on a fix. We are planning to continue the collaboration to help drive tool chain improvements.

4. RESOURCE REQUIREMENTS

The work described herein included contributions from SNL, LANL and ORNL.

SNL: Christian Trott, Dan Sunderland

ANL: Nevin Liber

ORNL: Damien Lebrun-Grandie

5. CONCLUSIONS AND FUTURE WORK

The Kokkos Support project executed a training bootcamp event for a mix of new Kokkos users and existing ones. Optimization work with existing Kokkos users let to improvements in Kokkos and initiated a collaboration on support for Aurora. The latter identified compiler bugs in the Intel toolchain which were reported to the vendor.

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 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'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.