Practices and Challenges of Software Development for a Performance Portable Ecosystem

Nathan Ellingwood and Sivasankaran Rajamanickam, Sandia National Laboratories
Unclassified Unlimited Release (UUR) SAND No. SAND2020-6606 C

Introduction

The Kokkos Ecosystem provides a parallel programming model (Kokkos Core), linear algebra library (Kokkos Kernels), and tool support (Kokkos Tools), enabling users, such as scientific computing and data analytics applications, to run codes efficiently on the available DOE super computing systems.

Performance portable software boosts productivity for application developers seeking to run their codes on these current and upcoming HPC architectures (by placing burden of low-level kernel implementation and performance on the library). Supporting the plethora of architectures and compilers employed by this diverse set of platforms, as well as interoperability requirements with applications, requires extensive testing and poses challenges on testing and maintenance of a stable software stack without crippling developers' efforts to contribute new capabilities. The goal of this white paper is to share current software practices and efforts toward addressing these challenges.

Practices

<u>Development workflow overview</u>
The kokkos, kokkos-kernels, and kokkos-tools Github repositories consist of active *main* and *develop* branches, and frozen tagged release branches.

- The main branch is default, production-ready code, and held frozen between releases.
- The develop branch tracks contributions by developers (bug fixes, new features, enhancements, etc.); completed features are submitted as pull requests for code review and acceptance testing.
- Github's issue tracker is used for communication regarding requests and code progress.
- New capabilities require usage documentation, performance benchmarks, and scripts for reproducibility of results.
- The release cycle is approximately three months, resulting in the merge of the develop branch into the main branch and a tagged release branch.

Training and Support

- Tutorial material: over 250 slides and 25 hands-on exercises teach parallel programming techniques, library and tool usage.

- Half-day/full-day trainings and multi-day bootcamps conducted periodically each year.

- Slack channels are actively monitored and provide a venue for questions, discussion, etc., ~ 350 members.

Automated Testing

As a performance-portable software library, the Kokkos Ecosystem supports a wide variety of compilers and versions, hardware, and library configurations dependent in part on the compiler and hardware combination available, including:

- Compilers: GCC (4.8.4 – 9); Clang (3.6-9); Intel (16-19); NVCC (9.0-11); IBM/XL (16.1.1)

- Backends: Serial, OpenMP, Pthreads, Cuda, HIP (Experimental), OpenMPTarget (Experimental), HPX

- Hardware: x86_64 (various), KNL, ARMv8 ThunderX2, NVIDIA (K80, P100, V100), AMD (Zen CPU, Vega MI50 and MI60)

Full testing coverage requires hours of testing across several computing systems. To balance developer productivity while preserving software quality and stability, testing of the code base consists of two stages:

Pull request acceptance testing

- Pull request testing is automated through use of Jenkins and triggered automatically when a new pull request is submitted to the repository through Github.

- A fixed subset from the full combinatorial explosion of the test coverage space selected to provide enough testing-breadth to give reasonable confidence that code changes pass full testing coverage with high likelihood; clang-format used to enforce consistency of style and readability across the code base.

- Final acceptance of a pull request requires code review by a team member. Code reviews help improve software quality and disseminate knowledge of code updates to members of the team.

Nightly testing

- Nightly testing is automated by Jenkins. Nightly tests significantly expand on pull request testing to fill the missing gaps in coverage. More than 200 nightly tests run on all the aforementioned platforms.

Release/Promotion Testing

Kokkos and KokkosKernels exist as standalone libraries available on Github, and are foundational packages within the Trilinos scientific software library (under the "data services" scope). A new release of the Kokkos Ecosystem requires extensive testing within Trilinos, providing additional robustness:

- We do not assume the testing suite is fully comprehensive – testing with Trilinos exposes corner cases that may have been missed in the testing suite.

- Trilinos testing assumed complex enough to provide coverage for customer applications.

Vendor Interactions

Kokkos and Kokkos Kernels have been adopted by several applications within the Department of Energy, academia and industry. As a result, the functionality developed here has also become a first "acceptance test" for programming model features and kernel needs when a new hardware is introduced.

- Kokkos has become part of the test suite for new versions of C++ compilers developed by different vendors.

- Kokkos Kernels has become the "reference implementation" for several linear algebra and graph kernels in vendor libraries. For example, the team level kernels, sparse matrix-matrix multiplication algorithms have been adopted in several vendor kernels.

- Kokkos Tools has become well integrated into vendor tools so Kokkos::View names or Kernel names appear correctly in vendor tools.

This continued collaboration with industry benefits the Kokkos users as their software will be ready for new architectures as soon as they are available to applications.

Disclaimer

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-NA0003525.