Implementing Differentiation of the Kokkos Framework in Clad

GSoC 2024 project by Atell Krasnopolski
Mentors: Vaibhav Thakkar, Vassil Vassilev, Petro Zarytskyi

Who am I?

Atell Yehor Krasnopolski

Bachelor Student @ University of Wuerzburg, Germany

Major: Mathematics

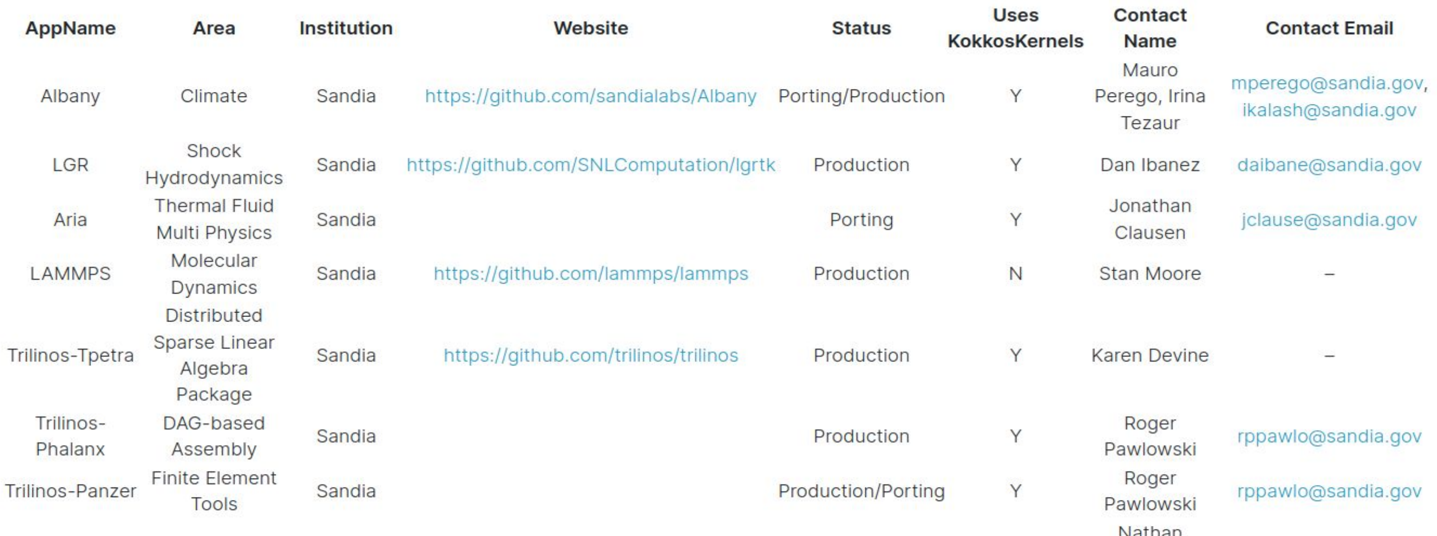
Experience: IRIS-HEP Fellow 2022, 2023

What is Kokkos

The Kokkos C++ Performance Portability Ecosystem is a production level solution for writing modern C++ applications in a hardware-agnostic way. It is part of the US Department of Energies Exascale Project – the leading effort in the US to prepare the HPC community for the next generation of supercomputing platforms

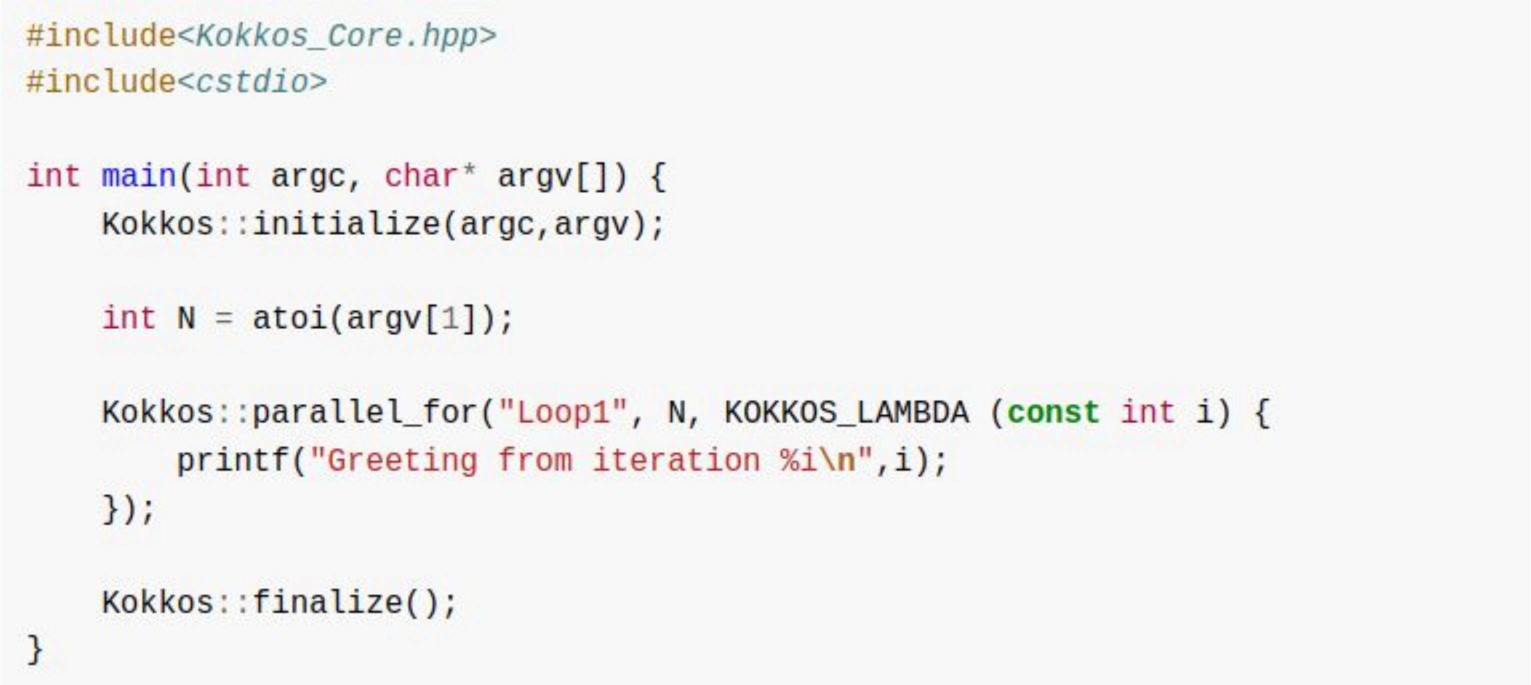
Kokkos applications

The Kokkos framework is used in several domains including climate modelling where gradients are an important part of the simulation process. This project aims at teaching Clad to differentiate Kokkos entities in a performance-portable way.



AppName	Area	Institution	Website	Status	Uses KokkosKernels	Contact Name	Contact Email
Albany	Climate	Sandia	https://github.com/sandialabs/Albany	Porting/Production	Y	Mauro Perego, Irina Tezaur	mperego@sandia.gov, ikalash@sandia.gov
LGR	Shock Hydrodynamics	Sandia	https://github.com/SNLComputation/lgrtk	Production	Y	Dan Ibanez	daibane@sandia.gov
Aria	Thermal Fluid Multi Physics	Sandia		Porting	Y	Jonathan Clausen	jclause@sandia.gov
LAMMPS	Molecular Dynamics	Sandia	https://github.com/lammps/lammps	Production	N	Stan Moore	–
Trilinos-Tpetra	Distributed Sparse Linear Algebra Package	Sandia	https://github.com/trilinos/trilinos	Production	Y	Karen Devine	–
Trilinos-Phalanx	DAG-based Assembly	Sandia		Production	Y	Roger Pawlowski	rppawlo@sandia.gov
Trilinos-Panzer	Finite Element Tools	Sandia		Production/Porting	Y	Roger Pawlowski	rppawlo@sandia.gov
						Nathan	

Kokkos basics



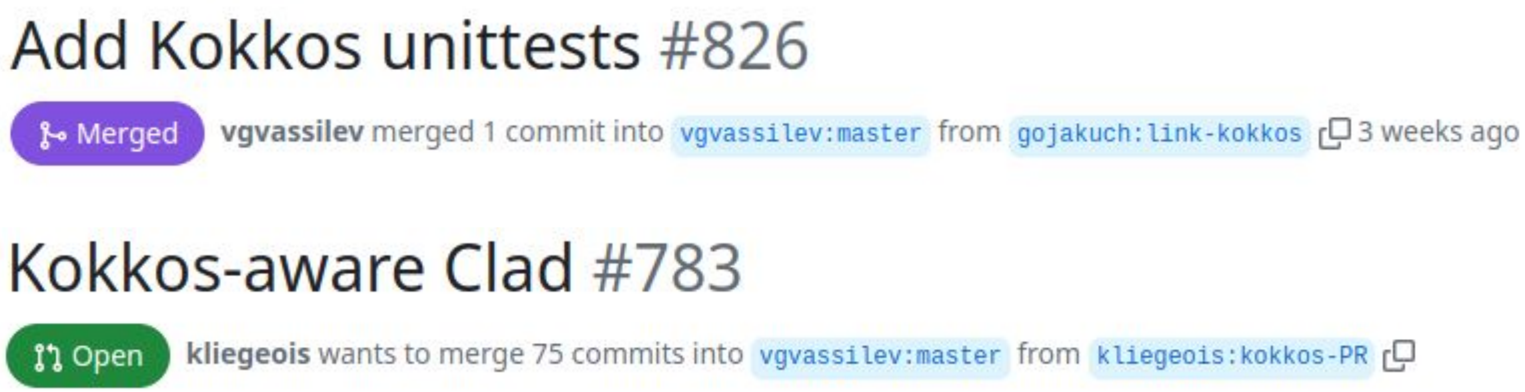
```
#include<Kokkos_Core.hpp>
#include<cstdio>

int main(int argc, char* argv[]) {
    Kokkos::initialize(argc,argv);

    int N = atoi(argv[1]);

    Kokkos::parallel_for("Loop1", N, KOKKOS_LAMBDA (const int i) {
        printf("Greeting from iteration %i\n",i);
    });

    Kokkos::finalize();
}
```

Goals (desired behaviour)

The goal is to implement the differentiation of the Kokkos high-performance computing framework including the support of:

- Kokkos functors,
- Kokkos lambdas,
- Kokkos methods such as parallel_for, parallel_reduce and deep_copy, as well as the general support for Kokkos::View data structures,
- Enhance existing benchmarks demonstrating effectiveness of Clad for Kokkos

Clad+Kokkos Backstory



Add Kokkos unittests #826

Merged vgvassilev merged 1 commit into `vgvassilev:master` from `gojakuch:link-kokkos` 3 weeks ago

Kokkos-aware Clad #783

Open kliegeois wants to merge 75 commits into `vgvassilev:master` from `kliegeois:kokkos-PR`

Generic approach

The approach should differ from what has been done previously to serve as a template.

Yet to be discussed.

Timeline

Full timeline available in the project proposal

- 01.05.2024-26.05.2024: Community bonding period
- 27.05.2024-Midterm evaluations: Main functionality in forward mode, view access in both modes
- Midterm evaluations-11.11.2024: Support for parallel_reduce and reverse mode differentiation, unit test enhancement, possibly support for other view-like data structures.

The timeline is planned to be extended until November 11th. The end date according to the initial plan was 25.08.2024.

Progress so far

- Kokkos unit tests (already mentioned)
- Most of the introductory steps done
- Currently working on issue #865 and familiarising myself with Clad