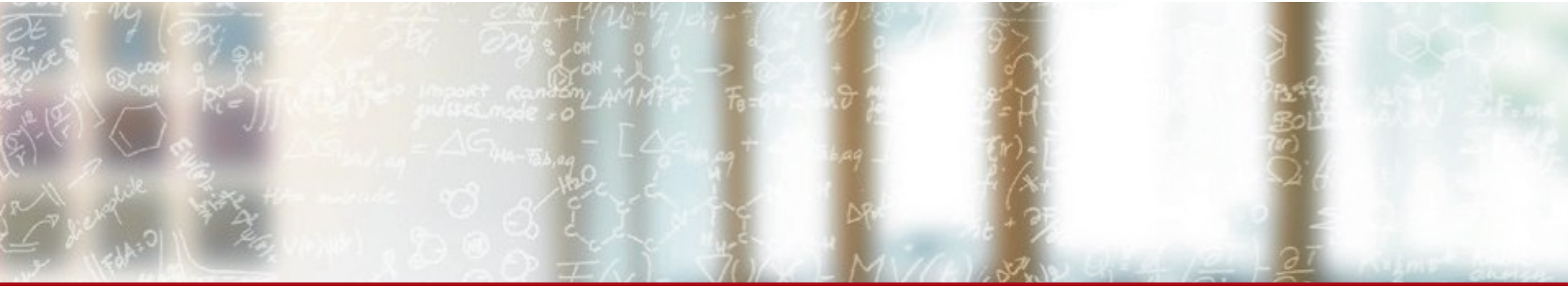
CSCS
Centro Svizzero di Calcolo Scientifico
Swiss National Supercomputing Centre

ETH zürich



Kokkos

CSCS User Lab Day

Mikael Simberg, CSCS

September 1, 2020

Table of Contents

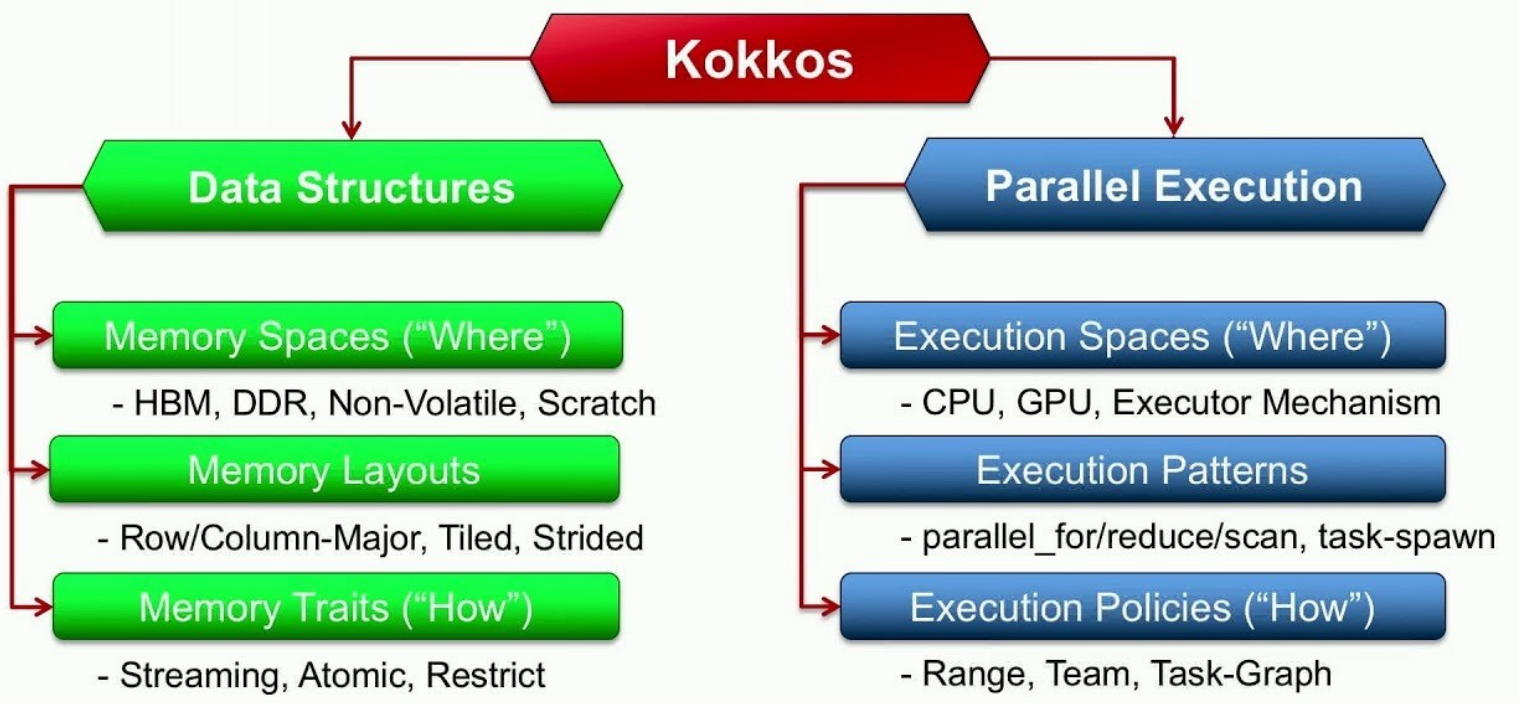
Motivation: Performance portability

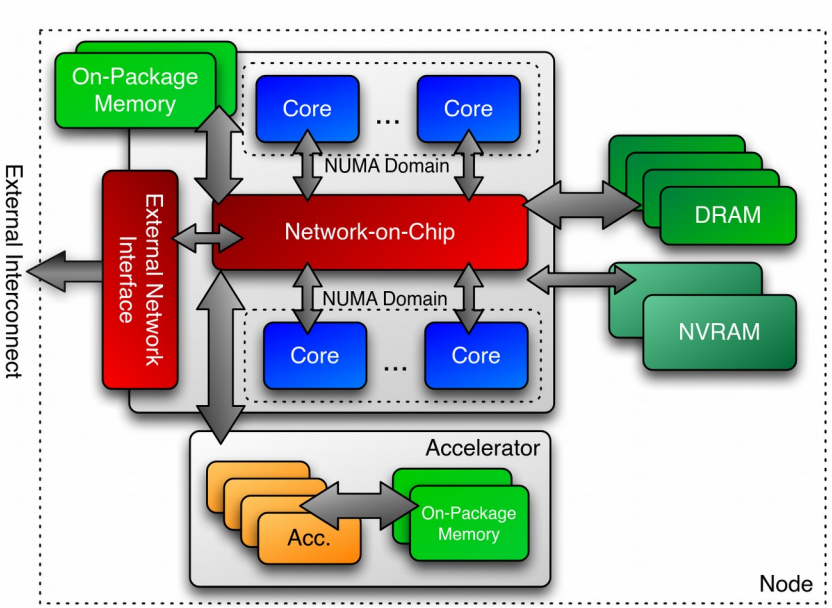
- Multiple major GPU vendors on the market, multiple major CPU vendors
  - All with slightly different runtimes and programming models
  - Competition good
  - Creates work for developers
- **Major rewrites expensive, so make them count**
  - Kokkos is an attempt to let you rewrite only once

What is Kokkos?

- Hierarchical data-parallelism, fine-grained tasking, and memory management on OpenMP, HPX, CUDA, HIP, SYCL etc. in the form of a pure C++ library
- **Single kernel implementation** for all backends
- *Not* a compiler, compiler extension, or runtime in itself
- Established
  - In development since 2012
  - Developed at Sandia National Labs, heavy investment by US DOE
- Large ecosystem built on top of Kokkos
  - Algorithms, tools, Trilinos

Kokkos abstractions





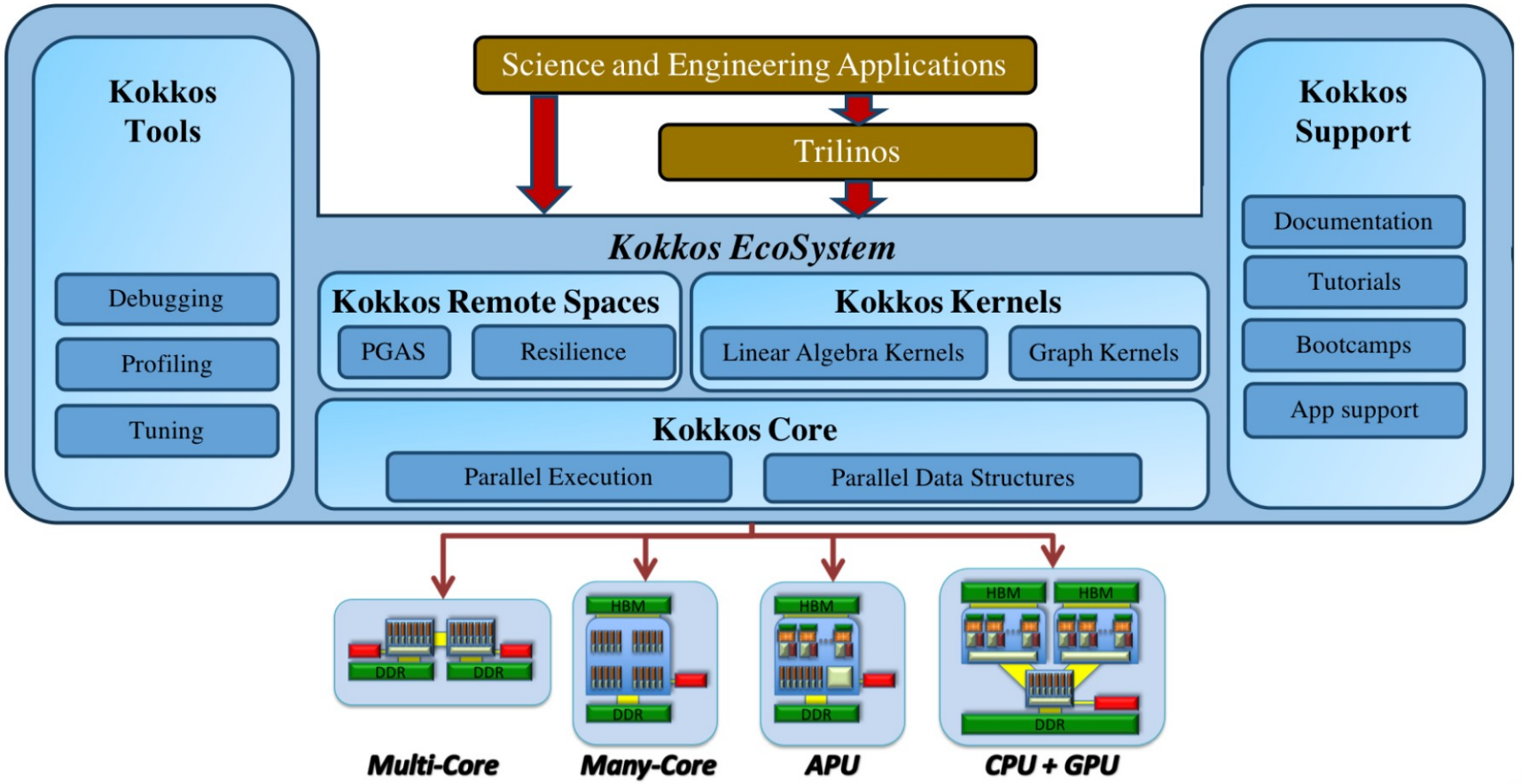
Credit: Kokkos Team

Example

```
Kokkos::View<int**> v(n, n);
Kokkos::parallel_for(
  MDRangePolicy<Kokkos::Rank<2>>({0, 0}, {n, n}),
  KOKKOS_LAMBDA(int i, int j) {
    v(i, j) = i * j;
  });
```

- View allocated on default device, reference counted
- Parallel for loop runs on default device
- View and MDRangePolicy use the appropriate access pattern for the device

The Kokkos ecosystem



Credit: Kokkos Team

CSCS
Centro Svizzero di Calcolo Scientifico
Swiss National Supercomputing Centre

ETH zürich

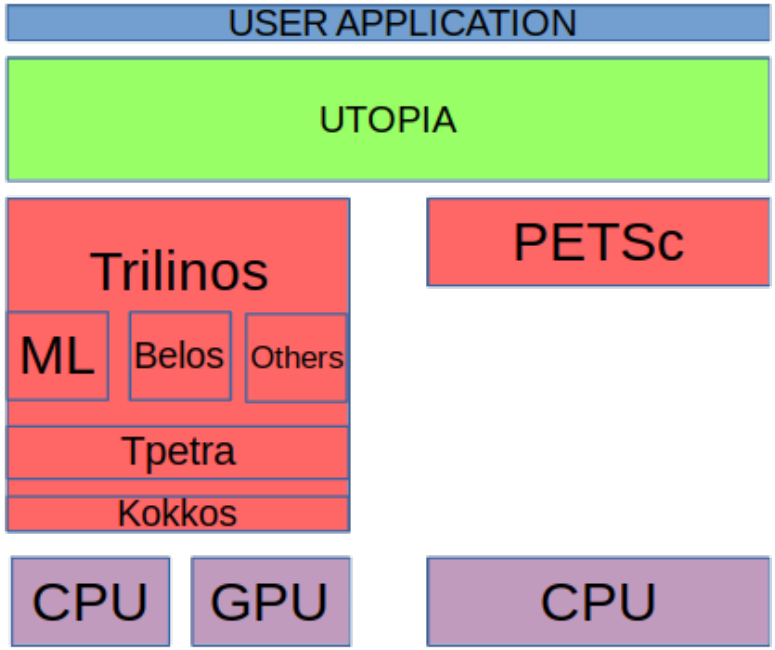
Kokkos at CSCS

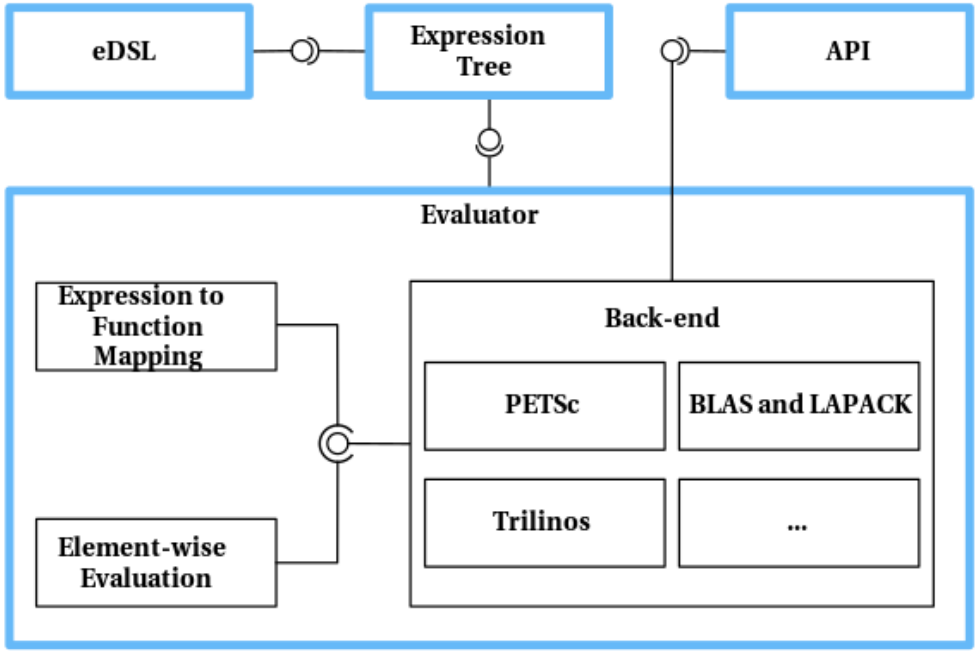
Applications and libraries using Kokkos at CSCS

- UTOPIA
  - Non-linear solvers built on top of Trilinos (and PETSc)
  - Nur Fadel and Andreas Fink at CSCS with Rolf Krause‘s group at USI
- MARS
  - Mesh generation on GPUs using pure Kokkos
  - Daniel Ganellari at CSCS together with Rolf Krause‘s group at USI
- SIRIUS
  - Electronic structure code which mixes Kokkos with e.g. normal CUDA code
  - Simon Pintarelli, Anton Kozhevnikov, and Mathieu Taillefumier at CSCS
- HPX
  - Interoperability with the HPX runtime, coarse grained tasking using HPX and Kokkos
  - John Biddiscombe, Auriane Reverdell, Mikael Simberg at CSCS together with LSU and Sandia

UTOPIA

- C++ expression templates for non-linear algebra
- Can lazily build trees of operations which are specialized for various backends
- **Trilinos backend (new) allows running on all backends supported by Kokkos**

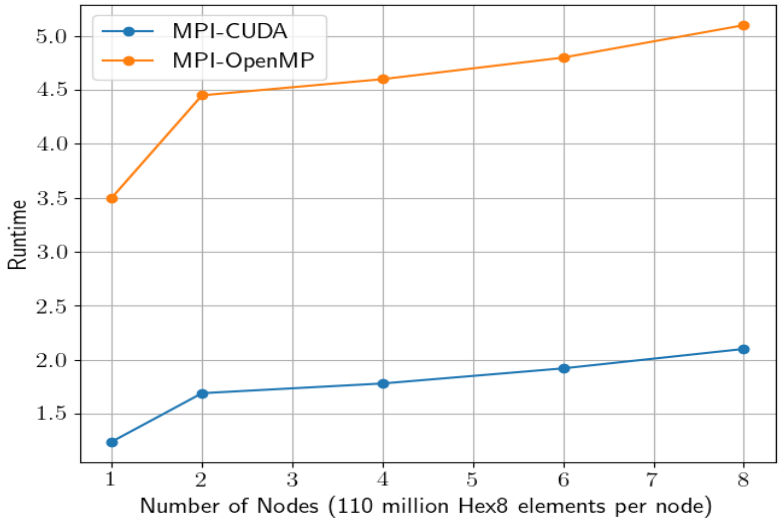


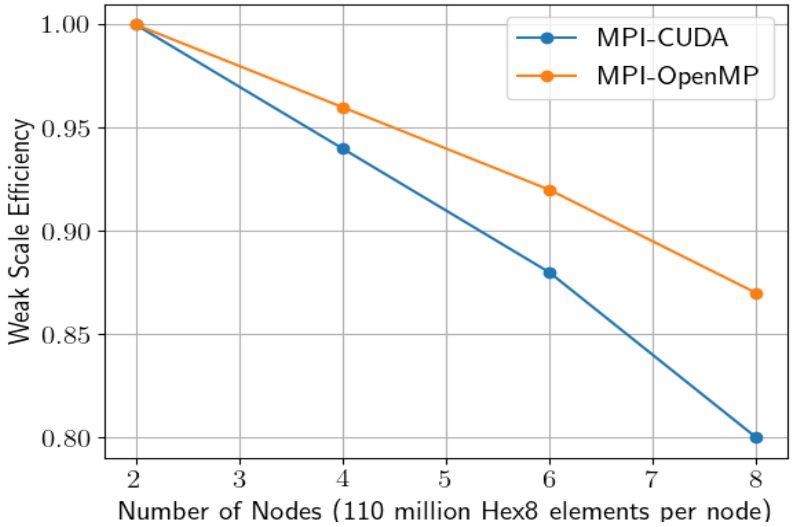


MARS

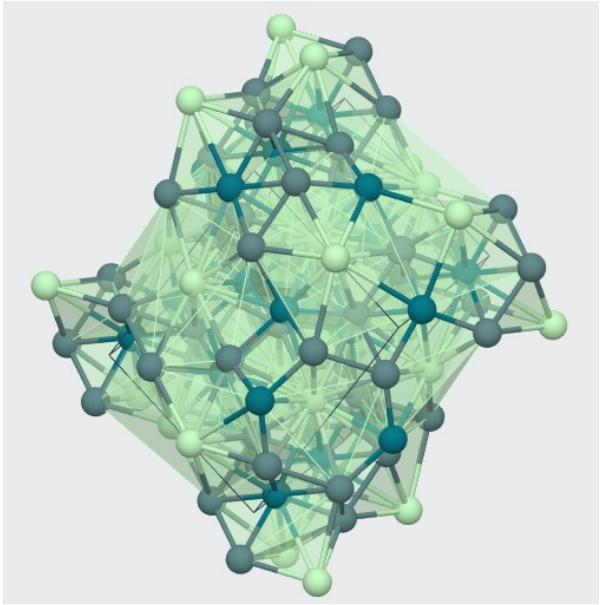
- Mesh Adaptive Refinement for Supercomputing
- C++ template meta-programming library using Kokkos
- **The mesh is entirely constructed on the device (GPUs)**
- Parallel mesh generation (3D) & AMR using LEPP (4D)
- Ex. 143 million Hex8 elements generated on a GPU node in only 0.86 sec
- Mesh management using Space Filling Curve algorithms, Morton z-curve

MPI-Kokkos (CUDA & OpenMP) Mesh Generation Weak Scaling and Efficiency





nlcglib: Non-linear CG algorithms for wave function optimization in ensemble DFT



- Solver plugin for SIRIUS and QuantumESPRESSO
- Contains wrappers for cublas, cusolver, BLAS/LAPACK
- Optimization coefficients stored in Kokkos views
- Unmanaged views to interface data with SIRIUS
- Certain operations performed with Kokkos parallel algorithms
- **Simple use of Kokkos, but shows importance of interoperability**

HPX

- HPX is closely aligned with the C++ standards for concurrency and parallelism
  - Provides a runtime with lightweight threads, async, futures, and parallel algorithms
  - Does not have good support for accelerators
- Interoperability
  - Kokkos has an HPX backend
  - Working on executors that forward to Kokkos execution spaces
- **Standardized and user-friendly API of HPX, portability of Kokkos**
  - HPX futures for synchronization
  - Kokkos kernels for bulk work
  - Combined give a way to create DAGs of operations, potentially increasing utilization of devices

Conclusion

Conclusion

- Kokkos is a C++ **performance portability** library: avoid having to rewrite algorithms and applications for every new runtime and architecture
- Used at CSCS both in applications and libraries
- Repository: https://github.com/kokkos/kokkos
- Kokkos lecture series a good in-depth resource (last lecture on Friday, but recordings and slides available): https://www.exascaleproject.org/event/kokkos-class-series/
- Questions?