

## [Introduction to Kokkos](#)

(the slides are available under "Presentation Materials" in the above URL)

Date: February 19, 2020

Presented by: Christian Trott (Sandia National Laboratories)

---

**Q.** Is the PGAS Space available for experimentation?

**A.** We are planning to make the repository available in March.

**Q.** How does labeling the operations help me?

**A.** The labels are used by profiling, debugging and tuning tools. They replace typeid's which are often really ugly in heavily templated code.

**Q.** Do kokkos/data structures provide persistence layer for HDF5 (parallel,serial)?

**A.** Not yet, but we have something in the works as a side project.

**Q.** Are OpenMP numbers for GPU or CPU?

**A.** GPU but it was a while ago.

**Q.** In the previous slide (15), when is " at this point" for XL OpenMP compiler? I think it is better now.

**A.** Yeah it was a while ago.

**Q.** Does kokkos insert omp SIMD when using thread vector range when executing on KNL?

**A.** It uses pragma ivdep for now, since about half our loops get slower when you force vectorization (omp simd makes the compiler ignore its cost model).

**Q.** When it is said that Kokkos is memory space aware, does it automatically utilize HBM or manage Linux HMM?

**A.** MemorySpaces are part of the typesystem. I.e. Kokkos::View is templated on the memory space, which means you can reason in your code about where data lives. We do not automatically transfer data, the user has to decide where to put things and when to move them.

**Q.** It seems Kokkos OpenMP backend are not portable across vendors. Why?

**A.** The CPU backend is fine, the GPU OpenMP Target backend is not super portable right now. One problem is that every vendor interpretes the standard differently, to the point that they have completely opposing views on which keywords should be ignored vs which ones have meaning. The other problem is that the compilers are not fully working. Every compiler except clang mainline (and with every compiler I also mean clang derivatives like AOMP from AMD) crashes with internal compiler errors when compiling our test suite.

**Q.** Clang works, IBM XL not working?

**A.** See above

**Q.** [github.com/kokkos/stdblas](https://github.com/kokkos/stdblas) gives a 404, is it private ?

**A.** Yes but going public very soon. For now the implementation is serial, it serves as a “yes the API works and is implementable” reference for the ISO C++ committee.

**Q.** Can someone outside the listed organizations join the Slack?

**A.** Yes, send me a message.

**Q.** Can kokkos be used for porting existing applications, implemented in C++ and also in fortran?

**A.** That has been the primary use case. I know of only a few from scratch new developments.

**Q.** How will the integration of Umpire Library look like?

**A.** We will expose Umpire through a templated memory space.

**Q.** Does kokkos provide Mac GPU support for mac os 10.14 and higher?

**A.** It should be possible to get it working on old macs with NVIDIA GPUs but we do not test that. We do not have an opencl backend which could be used on other integrated GPUs.

**Q.** Is there a plan to support multi-streaming for CUDA with kokkos ?

**A.** That is working. Contact us on SLACK for more details.

**Q.** It is not clear to me how/at what level kokkos integrates with MPI ? Which libraries are tested \* (openmpi vs mpich vs vendors ?) Ok so there is no like “grid” object possible across the whole cluster. Super, thanks!

**A.** Kokkos is mainly an on-node model like OpenMP or CUDA. It works with the same caveats with MPI (e.g. no MPI calls inside of parallel regions, you better make sure the MPI knows how to access the memory spaces you hand to the calls etc. )

**Q.** So there is no MPI-aware Kokkos. Are there any plans in this direction ?

**A.** We are working on the PGAS like abstractions, but we are not planning to introduce a message passing interface directly in Kokkos. MPI does the job fine.

**Q.** Any experience from using Kokkos with AMD CPUs + Nvidia GPUs; Are there any MPI Kokkos apps?

**A.** Almost all (>95%) of the applications using Kokkos also use MPI. And that includes obviously experience on AMD CPUs + NVIDIA GPUs.