Kokkos: Build Systems Update

Bruno Turcksin, Daniel Arndt, Damien Lebrun-Grandié
ORNL

Kokkos User Group Meeting 2023

December 12, 2023

Build Systems Supported

Build systems supported by Kokkos:

- CMake
- Makefile
- Spack
- Integration with Trilinos

Goal: Kokkos should always be installed the same way ⇒ this is difficult to do in practice

CMake

- Most tested build system
- Require version 3.16 or later
- Support installing the library
- Support inline build (copy Kokkos into your project and add add_subdirectory(kokkos) to your CMakeLists.tx)
- Configuration options always starts with **Kokkos_**

- Enabling backends:
  - at most the serial backend, one backend for multithreading on the host, and one device backend ⇒ not possible to enable SYCL and CUDA at the same time
  - if no host backend is enabled, the serial backend is enabled automatically
- Enabling architectures:
  - at most one CPU and one GPU
  - if a device backend is enabled but no architecture is provided, Kokkos tries to detect the architecture
  - **Kokkos_ARCH_NATIVE** optimize for the local CPU architecture

CMake CUDA

Every compilation unit containing Kokkos code has to be compiled with a CUDA-capable compiler:

- nvcc_wrapper: wrappers around nvcc to allow user to pretend that they have a GCC-compatible compiler
- CUDA Clang: behaves slightly different than nvcc
- nvc++: based on PGI (experimental)
- **Kokkos_ENABLE_COMPILE_AS_CMAKE_LANGUAGE**: downstream packages need to set the source file language

CMake CUDA

Recent changes in CUDA options:

- **Kokkos_ENABLE_CUDA_LAMBDA**: ON by default
- **Kokkos_ENABLE_CUDA_UVM**: deprecated

- Compilers supported: amdclang, hipcc, and Cray Clang
- Pick a compiler and stick to it
- Old naming scheme for AMD GPU made no sense: e.g. **Kokkos_ENABLE_VEGA90A** enables support of MI250 (Aldebaran)
- New naming scheme **Kokkos_ENABLE_AMD_GFX90A**. GFXYYY is the architecture flag passed to Clang
- We will continue to honor the old naming scheme for current architectures

- Naming scheme inspired AMD backend: e.g. **Kokkos_ENABLE_INTEL_GEN9**
- Still experimental

Makefile

- Not as thoroughly tested
- Only supports inline build
- We recommend using CMake

Spack

- Package maintained by the Kokkos Core and Kokkos Kernels teams
- Only a subset of options are available: use **spack info kokkos**
- Install Kokkos on A100: **spack install kokkos+cuda+wrapper cuda_arch=80**

Integration with Trilinos

- Kokkos snapshot in Trilinos ⇒ require compatibility with Trilinos
- Since Trilinos 14.2, no more Kokkos subpackages
- Since Trilinos 14.4 and Kokkos 4.1, Trilinos can use an external Kokkos
- Configure Trilinos using **cmake -DTPL_ENABLE_Kokkos=ON -DKokkos_DIR=/path/to/KokkosConfig.cmake ...**

QUESTIONS?