Accelerating Eigen Tensor libraries using SYCL

Mehdi Goli

University of West of Scotland, Codeplay Software Ltd.

Eigen is a high-level C++ library of template headers for linear algebra, matrix and vector operations, geometrical transformations, numerical solvers and related algorithms. The ecosystem of unsupported modules provides many specialized features such as tensor modules, matrix functions, a polynomial solver, FFT, and much more. We have implemented the tensor module back-end for SYCL™. SYCL is a royalty-free, cross-platform C++ abstraction layer that builds on the underlying concepts, portability and efficiency of OpenCL™, while adding the ease-of-use and flexibility of modern C++11/14. For example, SYCL enables single source development where C++ template functions are compiled for both host and device to construct complex algorithms that use OpenCL acceleration, and then re-use them throughout their source code on different types of data.

Implemented with ComputeCpp using the SYCL open standard, it is now possible to use the Eigen library with OpenCL devices, taking advantage of their ability to run operations in parallel using the expression templates meta-programming technique. Instead of rewriting each single operation for SYCL, we generate a compile-time DSL/EDSL that converts the existing Eigen expression to a SYCL based expression whenever it is possible. One of the advantages of using this non-intrusive approach is that both the Eigen SYCL backend and the CUDA backend use the same interface for the tensor module. This can be useful for higher-level existing applications like TensorFlow which automatically generates the Eigen expression for different back-ends. This approach eliminates the need for an extra expression generator in these applications for each back-end. Therefore, by templating the device it is possible to use the same expression generator approach for different back-end architectures.