The future direction of SYCL and C++ Heterogeneous Programming

Michael Wong[1]

[1] Codeplay Ltd., Edinburgh, United Kingdom

SYCL leads ISO C++ by supporting heterogeneous programming with modern C++. But when will ISO C++ catchup?

I would argue that heterogeneous support has been appearing in C++ since C++11, in bits and pieces. While there is no single TS or project that is named Heterogeneous, there is a quiet revolution to add support and I will highlight the C++ Parallel and Concurrency features that have driven in this direction.

In next week's C++ Standard meeting, C++ will soon ratify C++20 with the release of the Committee Draft. It will undoubtedly be the largest C++ release since C++11, with Concepts, Modules, Coroutines, Contracts all being major features. So, is there a deliberate future direction for C++? Or is it a random collection of features. I like to think there is with the establishment of a Direction Group.

SYCL is also expanding with vastly increased memberships, and many implementations of SYCL. The latest being the open source clang upstream led by Intel but collaborated across several companies, similar to the OpenMP effort I started many years ago to upstream clang and support offloading. It is a good time to look forward to see SYCL evolving with future C++ directions.