Min C++ Version in DPC++

oneAPI Language TAB, March 2020

SYCL Minimum C++ version

- The SYCL spec defines a **minimum** C++ version
  - Allows features from that C++ version to be used in the SYCL spec

- SYCL does **not** define a **maximum** C++ version
  - Use any features supported by an implementation (e.g. C++ experimental)
  - Only downside is lack of guaranteed portability
    - Minimum C++ version is guaranteed across SYCL implementations

DPC++ Minimum C++ Version == C++17

- Options:
  - Enable by default?
  - C++14++: Subset to C++17 features that are broadly supported (e.g. CTAD) and C++14 otherwise?
- Advantages:
  - Solid message of modern C++
  - CTAD for code simplicity
- Issues:
  - clang defaults to C++14, not C++17. Should compiler automatically bump to C++17?
  - Some customers don’t have host C++17 toolchains. But our primary tool supports
  - STL implementation probably lags. Don’t enable C++17 STL by default?