

# SUMMARY AND RECAP

## SYCL KEY CONCEPTS

- SYCL is a single-source, cross-platform abstraction C++ programming model
  - Device code and host code exist in the same file
  - Can target multiple heterogeneous APIs, such as OpenCL
  - Implementations enable these APIs by implementing SYCL backends
- SYCL provides high-level abstractions over common boilerplate code
  - Platform/device selection
    - Based on the OpenCL platform model
  - Buffer creation and data movement, USM
  - Dependency management and scheduling

# SYCL GPU BEST PRACTICES

- GPUs are massively parallel throughput devices
  - Important to give a GPU lots of work, thousands or millions of work-items
- Optimize Memory First
  - Monitor and minimize data transfer costs to or from the GPU
  - Improve performance by rearrange data or computation for locality
- Optimize Compute Next
  - Prefer smaller data types, trade precision for performance

## USEFUL LINKS

- [SYCL2020 Specification](#)
- [sycl.tech community](#)
- AdaptiveCPP:
  - <https://github.com/AdaptiveCpp/AdaptiveCpp>
  - [SYCL extensions](#)
- oneAPI Data Parallel C++ compiler:
  - <https://github.com/intel/llvm>
  - <https://tinyurl.com/dpcpp-ext>
- DPC++ books:
  - <https://tinyurl.com/dpcpp-book>
  - <http://tinyurl.com/dpcpp-book-2>
- Code Samples:
  - <https://github.com/oneapi-src/Velocity-Bench>
  - <https://github.com/oneapi-src/oneAPI-samples>

# QUESTIONS