

ONEAPI MATH KERNEL LIBRARY (ONEMKL) INTERFACES

LEARNING OBJECTIVES

- Learn about oneMKL library, more specifically oneMKL Interfaces project
- Learn about how to use GEMM APIs from oneMKL with both USM and buffer memory models

RESOURCES

- oneMKL Interfaces: <https://github.com/oneapi-src/oneMKL>
- oneMKL specification:
<https://spec.oneapi.io/versions/latest/elements/oneMKL/source/index.html#>
- Important: What is the difference between the following oneMKL terms: (1) oneAPI Specification for oneMKL (2) oneAPI's oneMKL Interfaces Project (3) Intel(R) oneAPI's oneMKL Product <https://github.com/oneapi-src/oneMKL?tab=readme-ov-file#onemkl>

RUN-TIME DISPATCHING

```
#include <oneapi/mkl/blas.hpp>

...

sycl::queue cpu_queue(sycl::cpu_selector_v);
sycl::queue gpu_queue(sycl::gpu_selector_v);

using namespace oneapi::mkl;
blas::column_major::gemm(cpu_queue,
                          transA, transB, m, ...);
blas::column_major::gemm(gpu_queue,
                          transA, transB, m, ...);
```

- Backend is loaded at run-time based on device-vendor
- `$> icpx -fsycl -I$ONEMKL/include app.cpp`
- `$> icpx -fsycl app.o -L$ONEMKL/lib -lonemkl`

COMPILE-TIME DISPATCHING

```
#include <oneapi/mkl/blas.hpp>

...

sycl::queue cpu_queue(sycl::cpu_selector_v);
sycl::queue gpu_queue(sycl::gpu_selector_v);

using namespace oneapi::mkl;
backend_selector<backend::mklcpu> cpu_selector(cpu_queue);
backend_selector<backend::cublas> gpu_selector(gpu_queue);

blas::column_major::gemm(cpu_selector,
                          transA, transB, m, ...);
blas::column_major::gemm(gpu_selector,
                          transA, transB, m, ...);
```

- Uses a templated backend selector APIs, where the template parameters specify the backends
- Application is linked with the required oneMKL backend wrapper libraries
- `$> clang++ -fsycl -I$ONEMKL/include app.cpp`
- `$> clang++ -fsycl app.o -L$ONEMKL/lib -lonemkl_blas_mklcpu -lonemkl_blas_cublas`

EXERCISE

- Objectives: Learn to use oneMKL GEMM buffer, USM APIs
- What is provided:
 - Boiler plate-code provided (a) to perform GEMM on CPU, (b) Helper function to verify results from oneMKL APIs and CPU
 - Please complete the TODO tasks marked in the source_*.cpp.
 - Refer to the solutions at solution_*.cpp