Intel® oneAPI Math Kernel Library (oneMKL)

Khang Nguyen
Agenda

- What is oneMKL
- What is new in oneMKL
- Running on CPU and GPU using dpc++
- Running on CPU and GPU using OpenMP Offload
- Building and linking
- Resources
- Q&A
What is oneMKL

- A set of highly optimized, threaded, and vectorized math functions to speed computations for scientific, engineering, and financial applications.
- Provides key functionality for dense and sparse linear algebra (BLAS, LAPACK, PARDISO), FFTs, vector math, summary statistics, splines, and more.
- Dispatches optimized code for each processor automatically without the need to branch code.
- Language support for DPC++ and Intel® C & Fortran compilers
- Full support for CPUs and select support for Intel® Processor Graphics Gen9+ and Intel® Xe GPUs
- Optimized for single-core vectorization and cache utilization
- Automatic parallelism for multi-core CPUs, GPUs, and scales from core to clusters
Inside oneMKL

Linear Algebra
- BLAS
- LAPACK
- ScaLAPACK
- Sparse BLAS
- Graph
- Cluster Sparse Solver
- PARDISO/Direct Sparse Solver

FFT Interfaces
- Multi-dimensional
- FFTW interfaces
- Cluster FFT

Vector RNGs
- Engines
- Distributions

Summary Statistics
- Kurtosis
- Variation coefficient
- Min/max
- Order Statistics
- Variance-covariance

Vector Math
- Trigonometric
- Hyperbolic
- Exponential
- Log
- Power
- Root

And More
- Splines
- Interpolation
- Trust Region
- Fast Poisson Solver

Intel® Processor Graphics Gen9/Gen12 & Intel Xe GPU support
Limited - Intel® Processor Graphics Gen9/Gen12 & Intel Xe GPU (see release notes)
What is New in oneMKL 2022

- **BLAS**
  - Added DPC++ support for in-place and out-of-place matrix copy/transposition.

- **LAPACK**
  - Enabled C/C++ OpenMP offload support for GETRI_OOP_BATCH.
  - Improved performance of double precision, non-pivoting batch strided LU factorization on GPU.
  - Improved performance of out-of-place batch strided LU inverse on GPU.
  - Renamed the LAPACK DPC++ function GETRFNP_BATCH_STRIDED to GETRFNP_BATCH.

- **Sparse**
  - Enabled C/C++ OpenMP offload support for MKL_SPARSE_SP2M and MKL_SPARSE_?_EXPORT_CSR.
  - Improved performance of DPC++ ONEAPI::MKL::SPARSE::MATMAT for small and medium sizes.
What is New in oneMKL 2022 - Continue

- **DFT**
  - Enabled MKL_VERBOSE support on GPU devices for DFT in DPC++ and C/C++/Fortran OpenMP offload.

- **Vector Math**
  - Performance and stability improvements

- **Library Engineering**
  - Enabled support for LP64 & ILP64 BLAS and LAPACK interfaces in a single application.

Example: DPC++ - Sine

**DPC++ API**

```cpp
using namespace cl::sycl;
constexpr size_t N = 256;
float A[N] = {0};

buffer<float, 1> A_buf{A, A+N};
buffer<float, 1> R_buf{N};

device dev{gpu_selector()};
queue q{dev};

oneapi::mkl::vm::sin(q, N, A_buf, R_buf, oneapi::mkl::vm::mode::la);
```

**C API**

```c
constexpr size_t N = 256;
float A[N] = {0}, R[N];

vmsSin(N, A, R, VML_LA);
```
Example: OpenMP Offload - DGEMM

double *A = ..., *B = ..., *C = ...;
const int dnum = 0;

#pragma omp target map(C[0:sizeC])
{
    Initialize C

#pragma omp target data map(to:A[...], B[...]) map(tofrom:C[...]) device(dnum)
{
    #pragma omp target variant dispatch device(dnum) use_device_ptr(A, B, C)
    {
        cblas_dgemm(CblasColMajor, CblasNoTrans, CblasNoTrans, m, n, k, alpha, A, lda,
                    B, ldb, beta, C, ldc);
    }
}
}
Building and Linking – DPC++ - Linux

- **Dynamic - Sequential**

  ```bash
  dpcpp -DMKL_ILP64 -I"${MKLROOT}/include" <your code>
  -L${MKLROOT}/lib/intel64 -lmkl_sycl -lmkl_intel_ilp64 -lmkl_sequential -lmkl_core -lsycl -lOpenCL -lpthread -lm -ldl -o <Executable file>
  ```

- **Static - TBB**

  ```bash
  dpcpp -DMKL_ILP64 -I"${MKLROOT}/include" <your code>
  -fsycl-device-code-split=per_kernel ${MKLROOT}/lib/intel64/libmkl_sycl.a -Wl,-export-dynamic -Wl,--start-group ${MKLROOT}/lib/intel64/libmkl_intel_ilp64.a
  ${MKLROOT}/lib/intel64/libmkl_tbb_thread.a ${MKLROOT}/lib/intel64/libmkl_core.a
  -Wl,--end-group -L${TBBROOT}/lib/intel64/gcc4.8 -ltbb -lsycl -lOpenCL -lpthread -lm -ldl -o <Executable file>
  ```

Note: dpc++ only supports ILP64
Building and Linking – C - OpenMP Offload - Linux

- Dynamic - OpenMP

```bash
icx -fiopenmp -fopenmp-targets=spir64 -DMKL_ILP64 -l"${MKLROOT}/include" <your code> -fsycl -L${MKLROOT}/lib/intel64 -lmkl_sycl -lmkl_intel_ilp64 -lmkl_intel_thread -lmkl_core -liomp5 -lsycl -lOpenCL -lstdc++ -lpthread -lm -ldl -o <Executable file>
```

- Static - OpenMP

```bash
icx -fiopenmp -fopenmp-targets=spir64 -DMKL_ILP64 -l"${MKLROOT}/include" <your code> -fsycl -Wl,--start-group ${MKLROOT}/lib/intel64/libmkl_sycl.a
${MKLROOT}/lib/intel64/libmkl_intel_ilp64.a
${MKLROOT}/lib/intel64/libmkl_intel_thread.a ${MKLROOT}/lib/intel64/libmkl_core.a -Wl,--end-group -liomp5 -lsycl -lOpenCL -lstdc++ -lpthread -lm -ldl -o <Executable file>
```
Building and Linking – Fortran - OpenMP Offload - Linux

- **Dynamic OpenMP**
  - `ifx -i8 -fiopenmp -fopenmp-targets=spir64 -DMKL_ILP64 -l"${MKLROOT}/include" -m64 -fpp <your code> -fsycl -L${MKLROOT}/lib/intel64 -lmkl_sycl -lmkl_intel_ilp64 -lmkl_intel_thread -lmkl_core -liomp5 -lsycl -lOpenCL -lstdc++ -lpthread -lm -ldl -o <executable file>

- **Static OpenMP**
  - `ifx -i8 -fiopenmp -fopenmp-targets=spir64 -DMKL_ILP64 -l"${MKLROOT}/include" -m64 -fpp <your code> -fsycl -Wl,--start-group ${MKLROOT}/lib/intel64/libmkl_sycl.a ${MKLROOT}/lib/intel64/libmkl_intel_ilp64.a ${MKLROOT}/lib/intel64/libmkl_intel_thread.a ${MKLROOT}/lib/intel64/libmkl_core.a -Wl,--end-group -liomp5 -lsycl -lOpenCL -lstdc++ -lpthread -lm -ldl -o <executable file>
## Resources

<table>
<thead>
<tr>
<th>Resource</th>
<th>URL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Intel® oneAPI Math Kernel Library (oneMKL)</td>
<td>software.intel.com/oneAPI/mkl</td>
</tr>
<tr>
<td>Intel® DevCloud for oneAPI</td>
<td>software.intel.com/en-us/devcloud/oneapi</td>
</tr>
<tr>
<td>Online Service Center (Paid Priority Support)</td>
<td>supporttickets.intel.com/servicecenter</td>
</tr>
<tr>
<td>oneMKL Benchmarks</td>
<td>software.intel.com/content/www/us/en/develop/tools/math-kernel-library/benchmarks.html</td>
</tr>
<tr>
<td>Intel® oneAPI MKL Link Line Advisor</td>
<td>software.intel.com/content/www/us/en/develop/articles/intel-mkl-link-line-advisor.html</td>
</tr>
<tr>
<td>Contacts</td>
<td><a href="mailto:mkl.tces@intel.com">mkl.tces@intel.com</a></td>
</tr>
</tbody>
</table>
Q&A