Intel® oneAPI Math Kernel Library (oneMKL)

Khang Nguyen

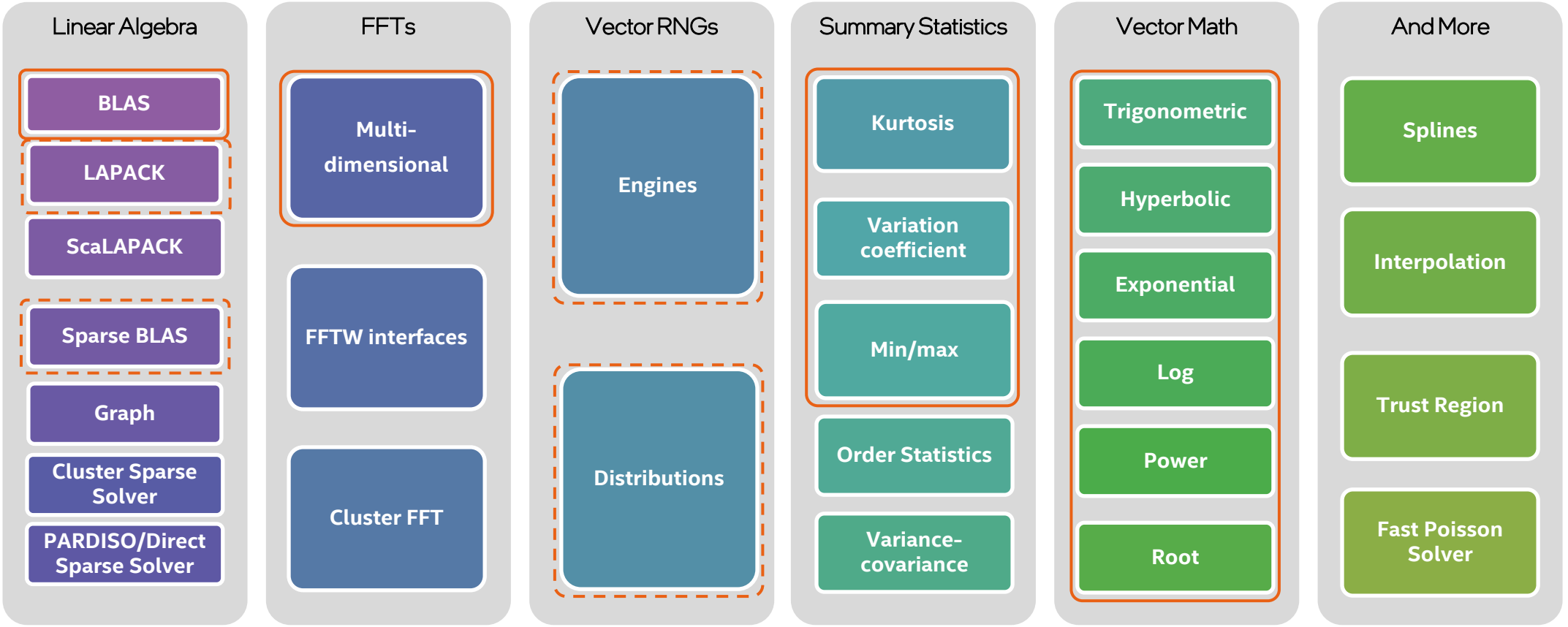
Agenda

- 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



Intel® Processor Graphics Gen9/Gen12 & Intel Xe GPU support

Limited - Intel® Processor Graphics Gen9/Gen12 & Intel Xe GPU (see release notes)

CPU C/Fortran support

What is New in oneMKL 2022

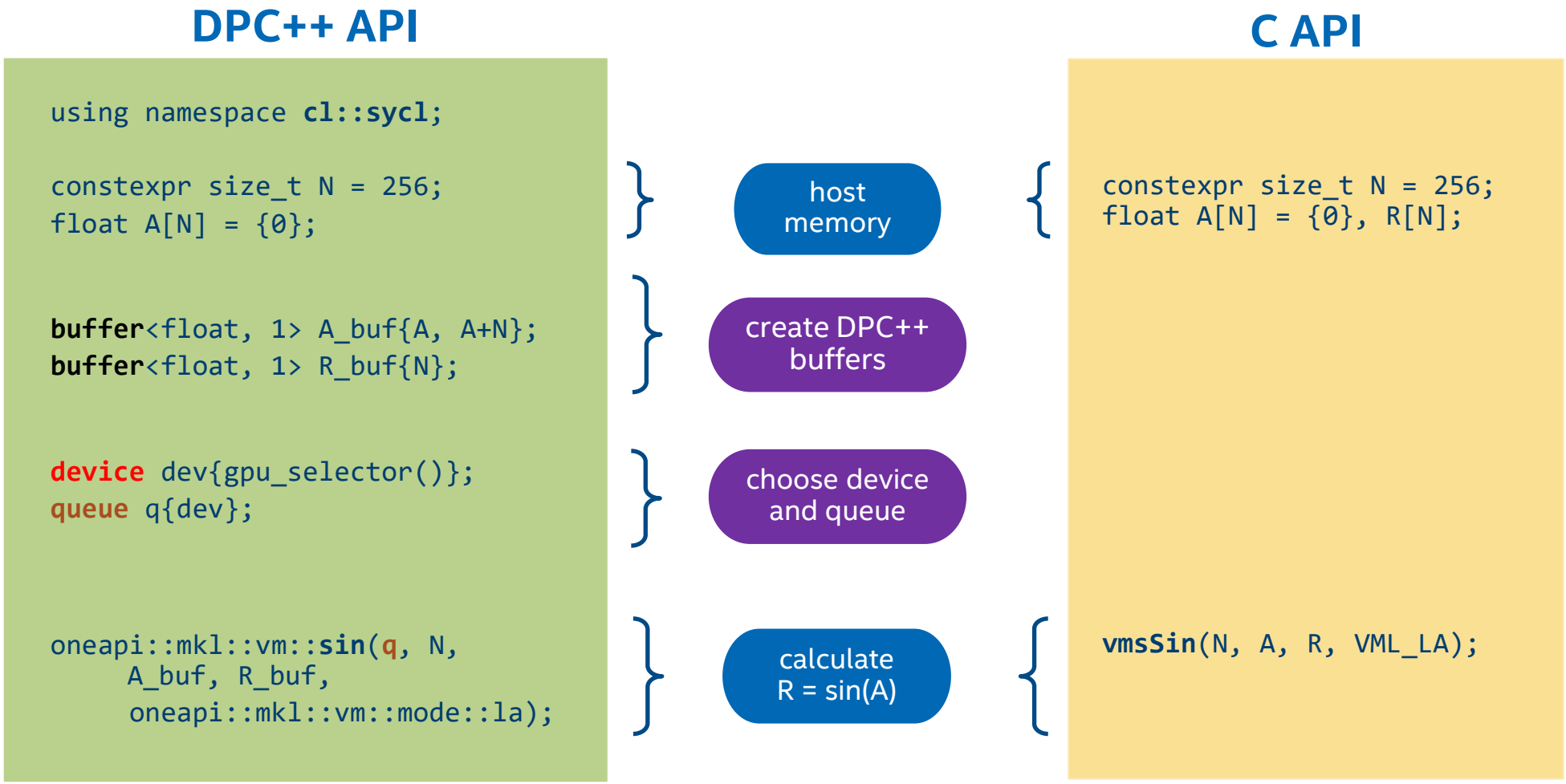
- BLAS
  - Added DPC++ support for in-place and out-of-place matrix copy/transposition.
- LAPACK
  - Enabled C/C++ OpenMP offload support for GETRI_OOP_BATCH.
  - Improved performance of double precision, non-pivoting batch strided LU factorization on GPU.
  - Improved performance of out-of-place batch 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.

https://software.intel.com/content/www/us/en/develop/articles/oneapi-math-kernel-library-release-notes.html

Example: DPC++ - Sine



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

```
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

```
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

icx -fiopenmp -fopenmp-targets=spir64 -DMKL_ILP64 -I"${MKLROOT}/include" <your code> -fsycl -L${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

icx -fiopenmp -fopenmp-targets=spir64 -DMKL_ILP64 -I"${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 -I"${MKLROOT}/include" -m64 -fpp <your code> -fsycl -L${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 -I"${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

Intel® oneAPI Math Kernel Library (oneMKL)	software.intel.com/oneAPI/mkl
Intel® DevCloud for oneAPI	software.intel.com/en-us/devcloud/oneapi
Online Service Center (Paid Priority Support)	supporttickets.intel.com/servicecenter
Intel Forum	community.intel.com/t5/Intel-oneAPI-Math-Kernel-Library/bd-p/oneapi-math-kernel-library
oneMKL Benchmarks	software.intel.com/content/www/us/en/develop/tools/math-kernel-library/benchmarks.html
Intel® oneAPI MKL Link Line Advisor	software.intel.com/content/www/us/en/develop/articles/intel-mkl-link-line-advisor.html
Contacts	mkl.tces@intel.com

Q&A


intel