

Intel[®] oneAPI Math Kernel Library (oneMKL)



October 2022

Gennady Fedorov, Technical Consulting Engineer, Intel Architecture, Graphics & Software (IAGS)

Intel® oneAPI Base Toolkit

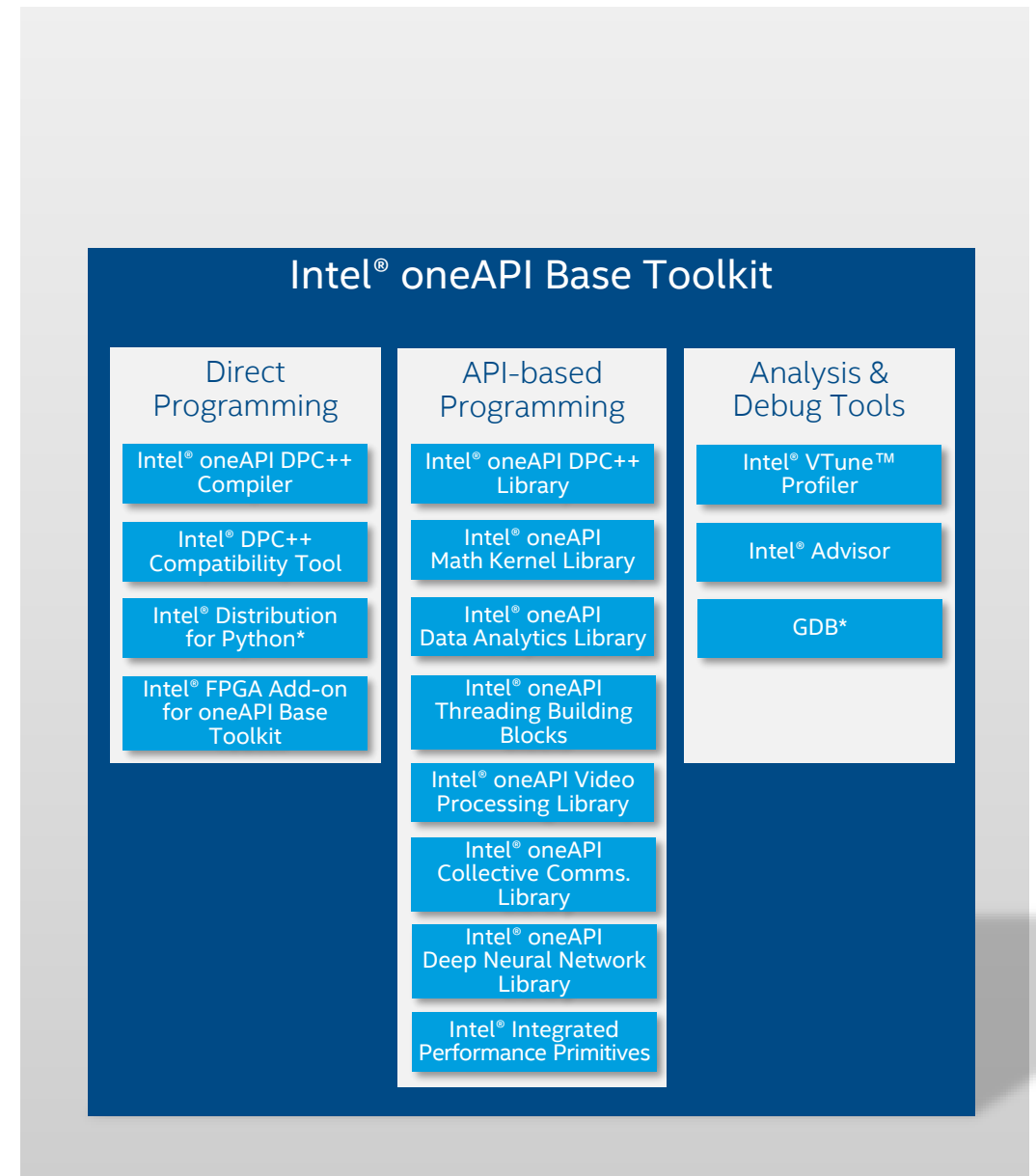
Core set of frequently used tools and libraries for developing high-performance applications across diverse architectures—CPU, GPU, FPGA.

Who Uses It?

- A broad range of developers across industries
- Add-on toolkit users because this is the base for all toolkits

Top Features/Benefits

- Data Parallel C++ (DPC++) compiler, library, and analysis tools
- DPC++ Compatibility tool helps migrate existing CUDA code
- Python distribution includes accelerated scikit-learn, NumPy, SciPy libraries
- Optimized performance libraries for threading, math, data analytics, deep learning, and video/image/signal processing



Intel® oneAPI Base Toolkit

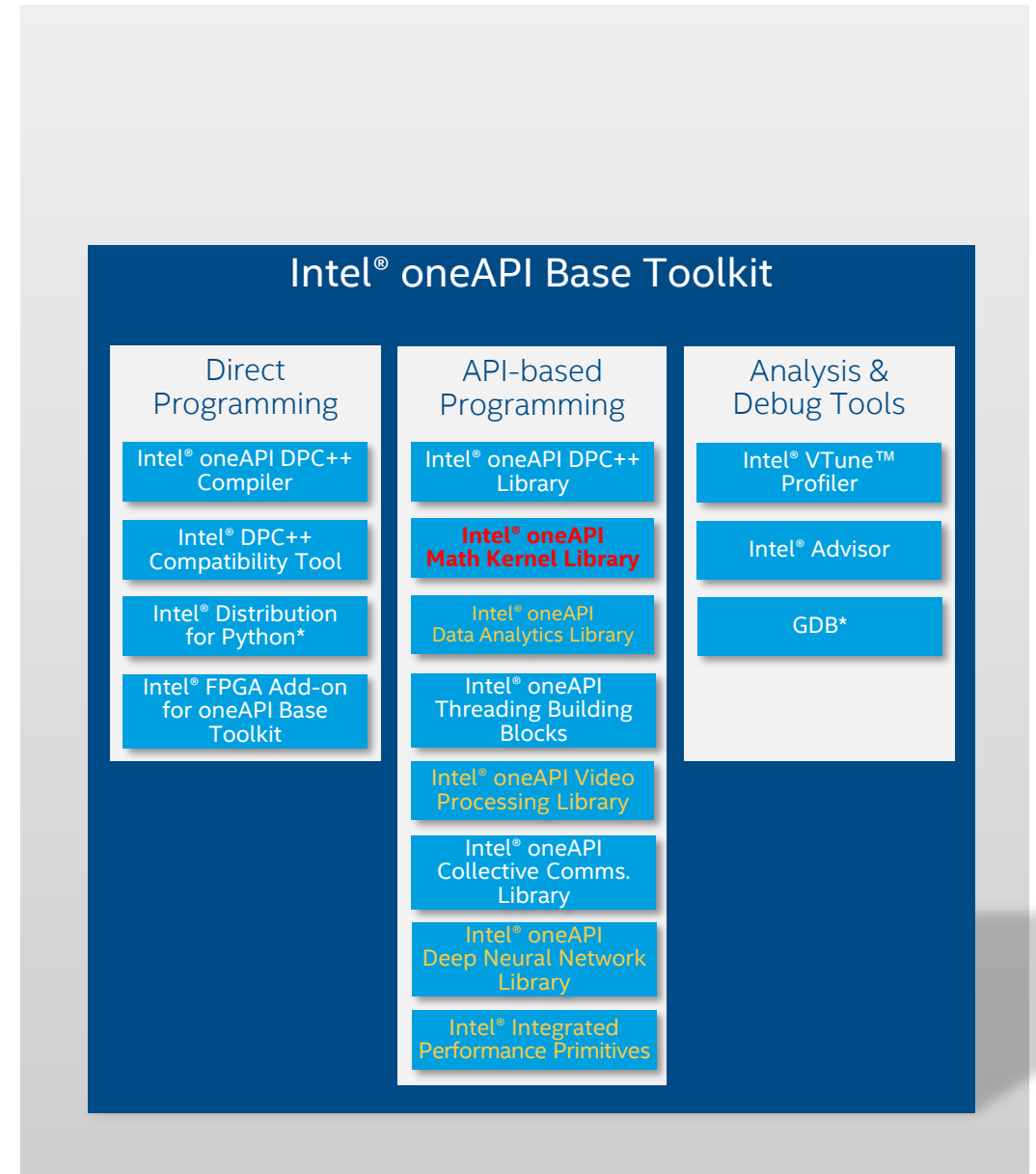
Core set of frequently used tools and libraries for developing high-performance applications across diverse architectures—CPU, GPU, FPGA.

Who Uses It?

- A broad range of developers across industries
- Add-on toolkit users because this is the base for all toolkits

Top Features/Benefits

- Data Parallel C++ (DPC++) compiler, library, and analysis tools
- DPC++ Compatibility tool helps migrate existing CUDA code
- Python distribution includes accelerated scikit-learn, NumPy, SciPy libraries
- Optimized performance libraries for threading, math, data analytics, deep learning, and video/image/signal processing



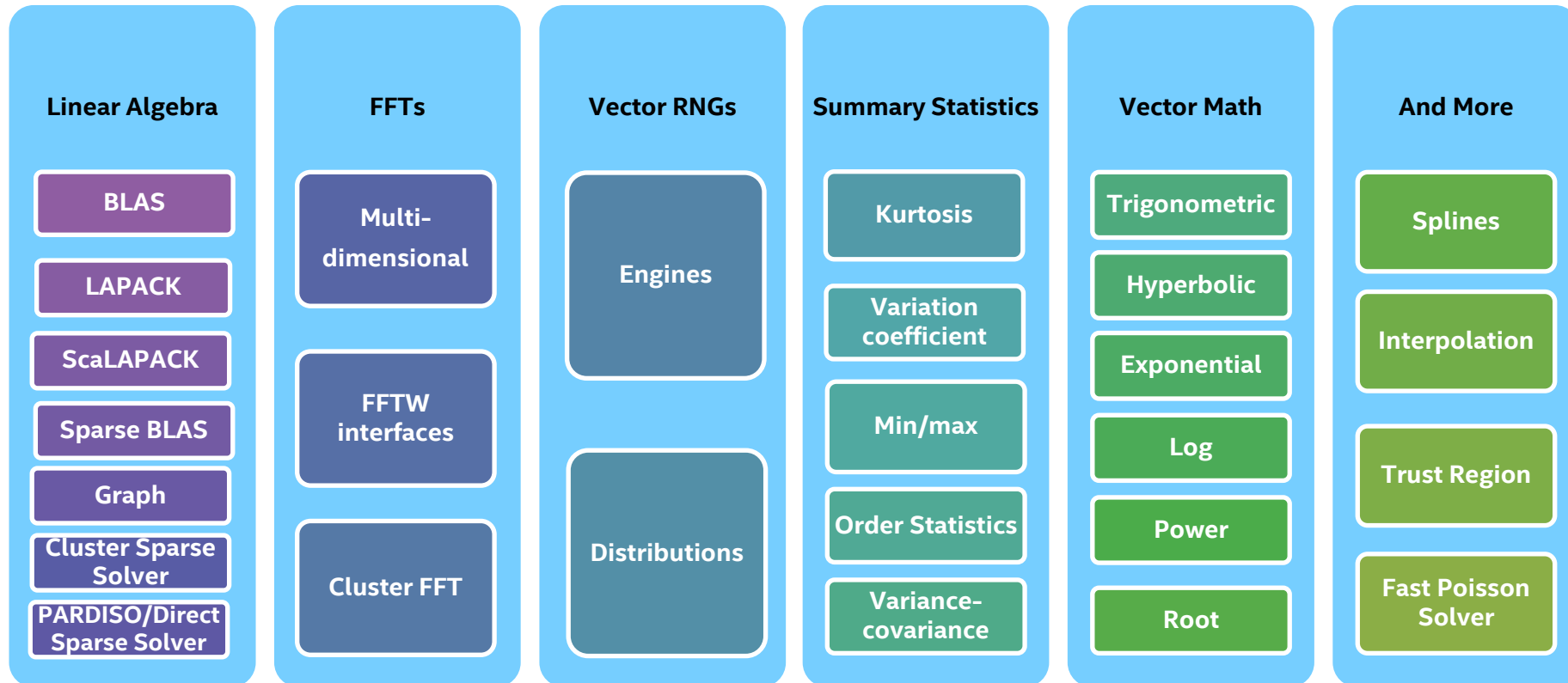
Agenda

- **What's New, Domain Area Updates**
- BLAS_64/Lapack_64 API Extensions
- MKL GPU Verbose mode
- Demo
- References

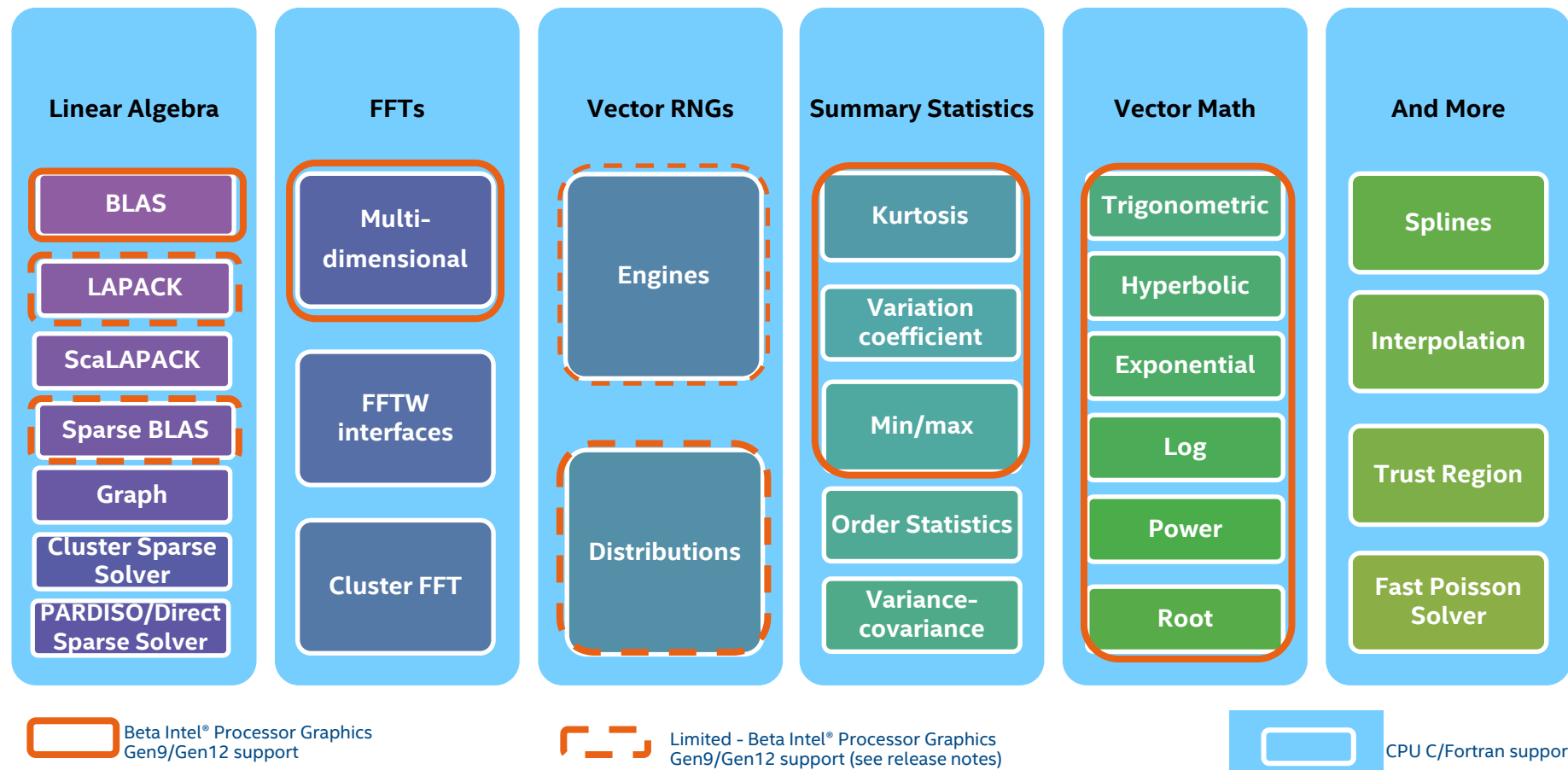
What's New for Intel® oneAPI MKL

- Data Parallel C++ (DPC++) APIs maximize performance and cross-architecture portability
- Introduces C and Fortran OpenMP offload for Intel® GPU acceleration
 - Support for Intel® Processor Graphics (GPU)
<https://software.intel.com/content/www/us/en/develop/articles/oneapi-math-kernel-library-system-requirements.html>
- oneAPI MKL Specification:
<https://spec.oneapi.com/versions/latest/elements/oneMKL/source/domains/domains.html>
- One MKL Open-Source interface: <https://github.com/oneapi-src/oneMKL>
- **Intel MKL continues to provide support for the same C and Fortran APIs for CPUs**

What's Inside Intel® MKL



Intel® oneAPI Math Kernel Library (oneMKL), cont



Intel® oneAPI MKL, BLAS, update

- BLAS, Netlib interfaces
 - USM and Buffer API - ALL
 - C/Fortran Offloading – ALL
 - BLAS Extensions

BLAS Level 1 Routines and Functions

- cblas_?asum
- cblas_?axpy
- cblas_?copy
- cblas_?dot
- cblas_?sdot
- cblas_?dotc
- cblas_?dotu
- cblas_?nrm2
- cblas_?rot
- cblas_?rotg
- cblas_?rotm
- cblas_?rotmg
- cblas_?scal
- cblas_?swap
- i?amax
- i?amin
- cblas_?cabs1

BLAS Level 2 Routines

- cblas_?gbmv
- cblas_?gemv
- cblas_?ger
- cblas_?gerc
- cblas_?geru
- cblas_?hbmw
- cblas_?hemv
- cblas_?her
- cblas_?her2
- cblas_?hpmv
- cblas_?hpr
- cblas_?hpr2
- cblas_?sbmv
- cblas_?spm
- cblas_?spr
- cblas_?spr2
- cblas_?symv
- cblas_?syr
- cblas_?syr2
- cblas_?tbmv
- cblas_?tbsv
- cblas_?tpmv
- cblas_?tpsv
- cblas_?trmv
- cblas_?trsv

BLAS Level 3 Routines

- cblas_?gemm
- cblas_?hemm
- cblas_?herk
- cblas_?her2k
- cblas_?symm
- cblas_?syrk
- cblas_?syr2k
- cblas_?trmm
- cblas_?trsm

OpenMP offload to support the OpenMP* 5.1 specification

Intel® oneAPI MKL, BLAS, update,

cont.

- BLAS, Netlib interfaces
 - USM and Buffer API - ALL
 - C/Fortran Offloading – ALL
 - **BLAS Extensions**

CPU	OpenMP Offload Intel GPU
{AXPY,GEMM,TRSM}_BATCH (group and strided)	{AXPY,GEMM,TRSM}_BATCH (group and strided)
GEMMT, AXPBY, GEMM3M	GEMMT
Integer GEMM (s8u8)	N/A
Bfloat16 GEMM	N/A
JIT GEMM API	N/A
PACK GEMM API	N/A
COMPACT GEMM API	N/A

Intel® oneAPI MKL, Sparse BLAS update

- Supported API:
 - gemm, gemv, trmv, trsv, symv
 - Buffer and USM API
 - C/C++ OpenMP Offloading (+ sp2m, mkl_sparse_?_mm)
- Improved performance of DPC++ `oneapi::mkl::sparse::matmat` for small to medium sizes
- Limitations:
 - CSR format only (CSC, COO, BSR, DIA and SKY) – no plans

Intel® oneAPI MKL, LAPACK update

DPC++ interfaces for selected routines from the Linear Algebra PACKage:

- Linear Equation Routines for solving, factoring, inverting tasks:
QR, LU, Bunch-Kaufman, Cholesky, GETRI, GETRS, POTRS, TRTRS...
- Singular Value and Eigenvalue Problem Routines:
GESVD, HEEVD, SYEVD, SYTRD, HEGVD, HETRD, ORGBR
- USM and Buffer API
- C/Fortran Offloading
- OpenMP offload to support the OpenMP* 5.1 specification

Intel® oneAPI MKL, LAPACK update, cont.

■ LAPACK Like Extensions:

- An additional routines to extend the functionality of the LAPACK routines. These include routines to compute many independent factorizations, linear equation solutions, and similar

- GEQRF_BATCH, GETRF_BATCH, GETRI_BATCH, GETRS_BATCH, ORGQR_BATCH, POTRF_BATCH,

POTRS_BATCH, UNGQR_BATCH

- <https://spec.oneapi.io/versions/latest/elements/oneMKL/source/domains/lapack/lapack.html>

- Buffer and USM API.

- C/Fortran Offloading (OpenMP offload to support the OpenMP* 5.1 specification)

Intel® oneAPI MKL, FFT update

- Supported modes:
 - Buffer and USM API
 - The same precision, domains, dimensions*, placement, batch, layout modes
 - C/Fortran Offloading – all FFT API (Level0 or OpenCL backends)

```
descriptor_t desc({N2, N1});
desc.set_value(oneapi::mkl::dft::config_param::NUMBER_OF_TRANSFORMS, BATCH);
desc.set_value(oneapi::mkl::dft::config_param::FWD_DISTANCE, N1*N2);
desc.set_value(oneapi::mkl::dft::config_param::BWD_DISTANCE, N1*N2);
desc.set_value(oneapi::mkl::dft::config_param::BACKWARD_SCALE, (1.0/(N1*N2)));
desc.commit(queue);
```

```
oneapi::mkl::dft::compute_forward(desc, inBuffer);
if (result == SUCCESS) {
    oneapi::mkl::dft::compute_backward(desc, inBuffer);
}
```

* 1,2 and 3 dimensions

config_param

```
enum class config_param {
    FORWARD_DOMAIN,
    DIMENSION,
    LENGTHS,
    PRECISION,

    FORWARD_SCALE,
    BACKWARD_SCALE,

    NUMBER_OF_TRANSFORMS,

    COMPLEX_STORAGE,
    REAL_STORAGE,
    CONJUGATE_EVEN_STORAGE,

    PLACEMENT,

    INPUT_STRIDES,
    OUTPUT_STRIDES,

    FWD_DISTANCE,
    BWD_DISTANCE,

    WORKSPACE,
    ORDERING,
    TRANSPOSE,
    PACKED_FORMAT,
    COMMIT_STATUS
};
```

Agenda

- What's New, Domain Area Updates
- **BLAS_64/Lapack_64 API Extensions**
- MKL GPU Verbose mode
- Demo
- References

BLAS_64/Lapack_64 API Extensions

1/4

- Using BLAS and LAPACK with the 32-bit and 64-bit interface (lp64 / ilp64) at the same time
- BLAS_64 and LAPACK_64 - NetLib interfaces
- Declaration: mkl_blas_64.h, mkl_lapack.h
- Limitations :
 - Intel64 only.
 - no Fortran API at this moment
 - ~~no mkl_lapacke.h (LAPACKE_cgetrf(*.....))~~ available since v.2022 update 2
 - CPU only

Agenda

- What's New, Domain Area Updates
- BLAS_64/Lapack_64 API Extensions
- **MKL GPU Verbose mode**
- Demo
- References

MKL Verbose mode

- CPU support
 - set/export MKL_VERBOSE=0*|1
 - mkl_verbose(int), mkl_verbose_output_file(char*)
- BLAS, FFT, LAPACK, ScaLAPACK
 - BLAS – no JIT, imatcopy, omatcopy...
 - ScaLAPACK:
 - P?POTRF, P?TRTRI, PDSYEV{D, R, X} and PZHEEV{D, R, X}.
 - All MPI ranks will print MKL_VERBOSE output.
 - RNG – TBD, VML – no plans
 - SpBLAS, Solvers

MKL Verbose mode, cont.

Examples:

MKL_VERBOSE oneMKL 2021.0 Update 4 Product build 20210904 for Intel(R)
64 architecture Intel(R) Advanced Vector Extensions 512 (Intel(R) AVX-512)
enabled processors, Lnx 2.20GHz lp64 intel_thread

MKL_VERBOSE

DGEMM(N,N,1280,1280,1280,0x7ffe1f04eb78,0x2b3062d68080,1280,0x2b306
39e9080,1280,0x7ffe1f04eb80,0x2b306466a080,1280) 7.09ms CNR:OFF Dyn:1
FastMM:1 TID:0 NThr:28

MKL_VERBOSE

FFT(**dcbi5x13x7**,tLim:22,desc:0xeb6d80) 93.11us CNR:OFF Dyn:1 FastMM:1 TID:0
NThr:44

MKL GPU Verbose mode, cont.

GPU Verbose mode – extension of the existing env variable and run time functions

To change the verbose mode, do one of the following:

- set the environment variable *MKL_VERBOSE*

	<i>CPU Targets</i>	<i>GPU Targets</i>
<i>(default) Set MKL_VERBOSE to 0</i>	to disable verbose	to disable verbose
<i>Set MKL_VERBOSE to 1</i>	to enable verbose	to enable verbose without timing
<i>Set MKL_VERBOSE to 2</i>	to enable verbose	to enable verbose with synchronous timer

- Or call the support function *mkl_verbose(int mode)*

	<i>CPU Targets</i>	<i>GPU Targets</i>
<i>(default) Call mkl_verbose(0)</i>	to disable verbose	to disable verbose
<i>Call mkl_verbose(1)</i>	to enable verbose	to enable verbose without timing
<i>Call mkl_verbose(2)</i>	to enable verbose	to enable verbose with synchronous timer

Intel® oneMKL Resources

Intel® oneMKL Product Page	https://www.intel.com/content/www/us/en/developer/tools/oneapi/onemkl.html						
Get Started with Intel® oneMKL	https://www.intel.com/content/www/us/en/develop/documentation/get-started-with-mkl-for-dpcpp/top.html						
Intel® oneMKL Developer Reference	https://www.intel.com/content/www/us/en/develop/documentation/onemkl-developer-reference-c/top.html						
Intel® oneMKL Developer Guide	https://www.intel.com/content/www/us/en/develop/documentation/onemkl-windows-developer-guide/top.html						
Intel® oneMKL Specification	https://spec.oneapi.io/versions/latest/elements/oneMKL/source/index.html						
Intel® oneMKL Open-Source Interface	https://github.com/oneapi-src/oneMKL						
Intel® oneMKL Release Notes	https://cqpreview.intel.com/content/www/us/en/developer/articles/release-notes/onemkl-release-notes.html						
Intel® oneMKL Forum	https://community.intel.com/t5/Intel-oneAPI-Math-Kernel-Library/bd-p/oneapi-math-kernel-library						

Notices & Disclaimers

Intel technologies may require enabled hardware, software or service activation. Learn more at intel.com or from the OEM or retailer.

Your costs and results may vary.

Intel does not control or audit third-party data. You should consult other sources to evaluate accuracy.

Optimization Notice: Intel's compilers may or may not optimize to the same degree for non-Intel microprocessors for optimizations that are not unique to Intel microprocessors. These optimizations include SSE2, SSE3, and SSSE3 instruction sets and other optimizations. Intel does not guarantee the availability, functionality, or effectiveness of any optimization on microprocessors not manufactured by Intel. Microprocessor-dependent optimizations in this product are intended for use with Intel microprocessors. Certain optimizations not specific to Intel microarchitecture are reserved for Intel microprocessors. Please refer to the applicable product User and Reference Guides for more information regarding the specific instruction sets covered by this notice. Notice Revision #20110804. <https://software.intel.com/en-us/articles/optimization-notice>

Software and workloads used in performance tests may have been optimized for performance only on Intel microprocessors.

Performance tests, such as SYSmark and MobileMark, are measured using specific computer systems, components, software, operations and functions. Any change to any of those factors may cause the results to vary. You should consult other information and performance tests to assist you in fully evaluating your contemplated purchases, including the performance of that product when combined with other products. See backup for configuration details. For more complete information about performance and benchmark results, visit www.intel.com/benchmarks.

Performance results are based on testing as of dates shown in configurations and may not reflect all publicly available updates. See configuration disclosure for details. No product or component can be absolutely secure.

No license (express or implied, by estoppel or otherwise) to any intellectual property rights is granted by this document.

Intel disclaims all express and implied warranties, including without limitation, the implied warranties of merchantability, fitness for a particular purpose, and non-infringement, as well as any warranty arising from course of performance, course of dealing, or usage in trade.

© Intel Corporation. Intel, the Intel logo, and other Intel marks are trademarks of Intel Corporation or its subsidiaries. Other names and brands may be claimed as the property of others.

intel®