INTEL® MATH KERNEL LIBRARY, BLAS USAGE

Gennady.Fedorov@intel.com

What's Inside Intel® MKL

Accelerate HPC, Enterprise, IoT & Cloud Applications

Linear Algebra
- **BLAS**
- LAPACK
- ScaLAPACK
- Sparse BLAS
- Iterative sparse solvers
- PARDISO*
- Cluster Sparse Solver

FFTs
- Multidimensional
- FFTW interfaces
- Cluster FFT

Neural Networks
- Convolution
- Pooling
- Normalization
- ReLU
- Inner Product

Vector RNGs
- Congruential
- Wichmann-Hill
- Mersenne Twister
- Sobol
- Neiderreiter
- Non-deterministic

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

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

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

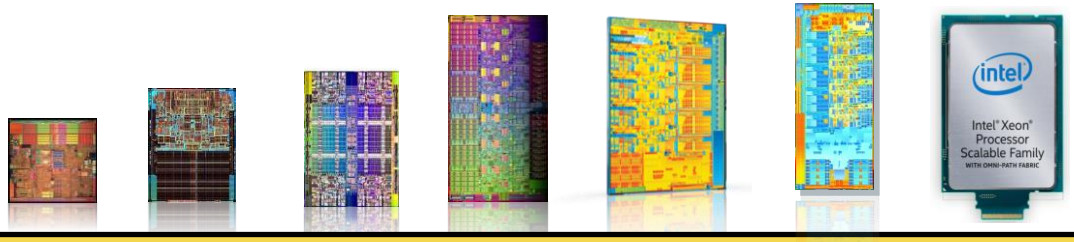
Intel® Architecture Platforms

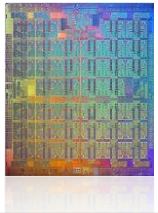
Operating System: Windows*, Linux*, MacOS[1]*

[1] Available only in Intel® Parallel Studio Composer Edition.

Automatic Dispatching to Tuned ISA-specific Code Paths

More cores → More Threads → Wider vectors





	Intel® Xeon® Processor 64-bit	Intel® Xeon® Processor 5100 series	Intel® Xeon® Processor 5500 series	Intel® Xeon® Processor 5600 series	Intel® Xeon® Processor E5-2600 v2 series	Intel® Xeon® Processor E5-2600 v3 series v4 series	Intel® Xeon® Scalable Processor[1]	Intel® Xeon Phi™ x200 Processor (KNL)
Up to Core(s)	1	2	4	6	12	18-22	28	72
Up to Threads	2	2	8	12	24	36-44	56	288
SIMD Width	128	128	128	128	256	256	512	512
Vector ISA	Intel® SSE3	Intel® SSE3	Intel® SSE4- 4.1	Intel® SSE 4.2	Intel® AVX	Intel® AVX2	Intel® AVX-512	Intel® AVX-512

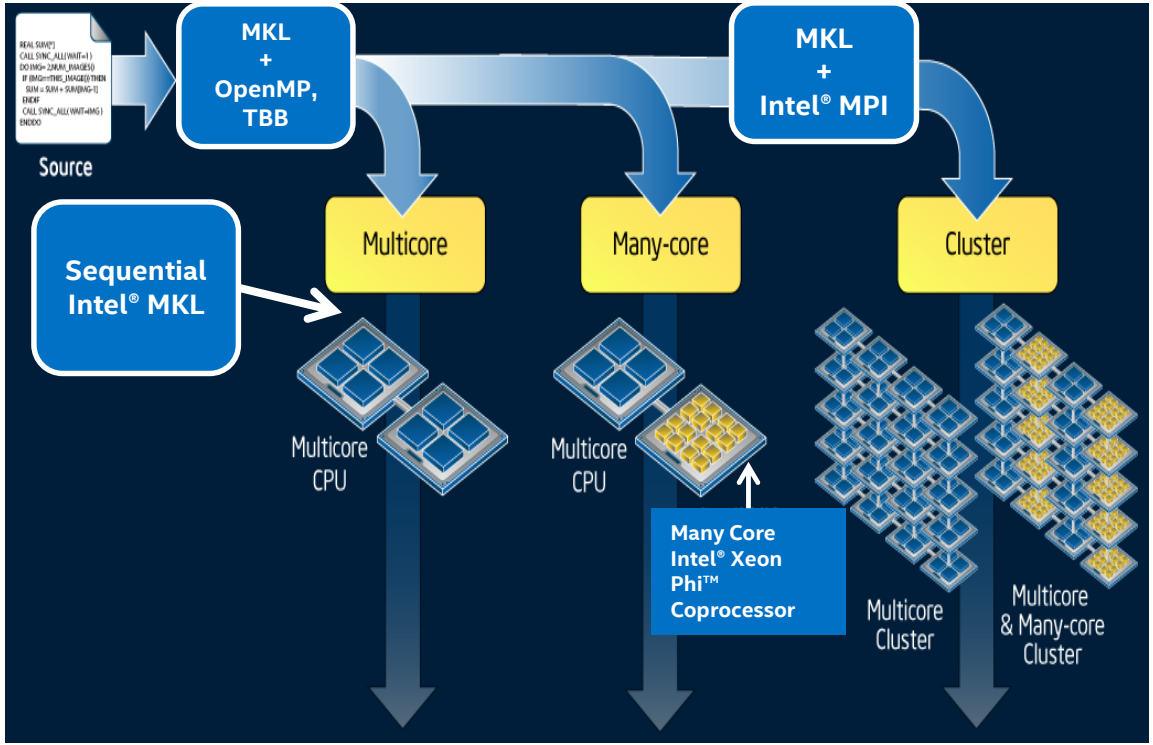
1. Product specification for launched and shipped products available on ark.intel.com.

Automatic Performance Scaling from the Core, to Multicore, to Many Core and Beyond

Intel® MKL

Extracting performance from the computing resources

- Core: **vectorization**, prefetching, cache utilization
- Multi-Many core (processor/socket) level **parallelization**
- Multi-socket (node) level **parallelization**
- Clusters **scaling**



Intel® MKL 2018 New Features and Optimizations

Intel® Xeon Phi™ Knights Mill Optimizations	DNN Convolution and Inner Product functions Single Precision Level 3 BLAS, including SGEMM New INT8 and INT16 GEMM functions
BLAS and LAPACK	Compact BLAS and LAPACK functions Direct Call LAPACK Cholesky and QR factorizations LU factorization and Inverse without pivoting Aasen-based factorization and solve functions Bounded Bunch-Kaufman (Rook) pivoting factorizations
FFTs	Verbose Mode Support
Vector Math - 24 New Functions	v?Fmod, v?Remainder v?Powr, v?Exp2, v?Exp10 v?Cospi, v?Sinpi, v?Tanpi, and more

Introduction

- Objective:
  - In these activities, you will run Intel® MKL DGEMM based code
  - Learn how to manage the output and performance
- Requirements:
  - Intel® Parallel Studio XE 2018 Composer Edition with Intel® C++ Compiler
  - **Linux*** OS supported by Intel® C++ Compiler
  - Recommended to have at least 3nd **generation Intel® Core™ processor** (with Intel® AVX2)
- **Time : 25 min**

Intel® MKL Lab – BLAS Usage **Activity #1**

Folder **day1/lab2 :**

mkl_test_v1.c, mkl_test_v2.c, mkl_test_v3.c

- review test: cat/less **mkl_test_v1.c.**
  - mkl_malloc, mkl_free, dsecnd,
  - roll_your_own_multiply,
  - Ddot_Multiply,
  - Dgemv_multiply,
  - Dgemm_multiply

Note – “https://software.intel.com/en-us/articles/a-simple-example-to-measure-the-performance-of-an-intel-mkl-function”

Intel® MKL Lab – BLAS Usage **Activity #1**

GEMM API: **C = alpha*op(A)*op(B) + beta*C**

$C_{m,n} = A_{m.k} * B_{k,n}$

```
void cblas_dgemm (
const CBLAS_LAYOUT Layout, const CBLAS_TRANSPOSE transa, const CBLAS_TRANSPOSE transb,
const MKL_INT m, const MKL_INT n, const MKL_INT k,
const double alpha, const double *a,
const MKL_INT lda, const double *b, const MKL_INT ldb,
const double beta, double *c, const MKL_INT ldc );
```

Intel® MKL Lab – BLAS Usage **Activity #1**

#Compiling and Linking:

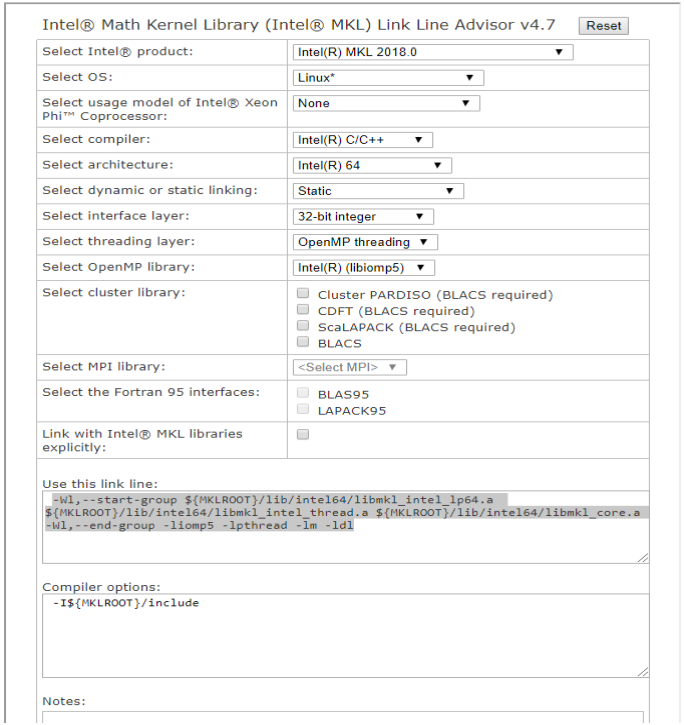
- Set compiler's environment:

source opt/intel/compilers_and_libraries_2018/linux/bin/**compilervars.sh intel64**

- **icc -mkl mkl_test_v1.c -o 1.out**

#MKL Linker Adviser :

https://software.intel.com/en-us/articles/intel-mkl-link-line-advisor



Intel® MKL Lab – BLAS Usage **Activity #1**

#Run and record execution times:

- ./1.out 1000
- ./1.out 2000
- ~~./1.out 4000~~

./1.out 1000

roll_your_own_multiply(). Elapsed time = ..., sec

Ddot_Multiply(). Elapsed time = ..., sec

Dgemv_multiply(). Elapsed time = ..., sec

Dgemm_multiply(). Elapsed time = ...,sec

Conclusion?

Intel® MKL Lab – BLAS Usage **Activity #2**

#review test: cat/less **mkl_test_v2.c**

- icc -mkl mkl_test_v2.c -o 2.out

#SCALABLITY: Run and record execution times

- export MKL_NUM_THREADS=1
  - ./2.out 8000 … Elapsed time = ….., sec
- export MKL_NUM_THREADS=2
  - ./2.out 8000 …. Elapsed time = ….., sec
- export MKL_NUM_THREADS=4
  - ./2.out 8000 …..Elapsed time = ….., sec

Conclusion?

Intel® MKL Lab – BLAS Usage **Activity #2**

mkl_test_v2.c : cblas_dgemm(….);

#Verbose Mode:

- export MKL_VERBOSE=1 // 0 by default

int mkl_verbose (true/false); // false by default

- **./2.out 8000**

MKL_VERBOSE Intel(R) MKL 2018.0 Update 2 Product build 20180127 for Intel(R) 64 architecture Intel(R) Advanced Vector Extensions 512 (Intel(R) AVX-512) enabled processors, Lnx 2.30GHz lp64 intel_thread

MKL_VERBOSE
DGEMM(N,N,8000,8000,8000,0x7fffbd682890,0x7f6d8a634080,8000,0x7f6d6bdeb080,8000,0x7fffbd682898,0x7f6d4d5a2080,8000) 4.65s CNR:OFF Dyn:1 FastMM:1 TID:0 NThr:4

Intel® MKL Lab – BLAS Usage **Activity #3**

#IA Dispatching

review test: cat/less **mkl_test_v3.c** : cblas_dgemm(….);

#Environment Variable:

MKL_ENABLE_INSTRUCTIONS == AVX512 | AVX2 | AVX | SSE4_2

#Runtime function:

int mkl_enable_instructions (int isa);

isa == MKL_ENABLE_AVX512 | MKL_ENABLE_AVX2 | MKL_ENABLE_AVX | MKL_ENABLE_SSE4_2

Intel® MKL Lab – BLAS Usage **Activity #3**

#IA Dispatching

mkl_test_v3.c : cblas_dgemm(….);

- icc -mkl mkl_test_v3.c -o 3.out
- export MKL_VERBOSE=1

#and run and record execution times:

- **./3.out 8000**
  - export MKL_ENABLE_INSTRUCTIONS=AVX2 … Elapsed time = ……, sec
  - export MKL_ENABLE_INSTRUCTIONS=AVX … Elapsed time = ……, sec
  - export MKL_ENABLE_INSTRUCTIONS=SSE4_2 …. Elapsed time = ……, sec
  - export MKL_ENABLE_INSTRUCTIONS=AVX512 … Elapsed time = ……., sec

Intel® MKL Lab – BLAS Usage **Activity #4**

#Conditional Numerical Reproducibility - CNR

mkl_test_v3.c

export MKL_ENABLE_INSTRUCTIONS=

export MKL_CBWR=**VALUE**

Value	Description
COMPATIBLE	Intel® Strea Intel® Streaming SIMD Extensions 2 (Intel® SSE2) without rcp
SSE2	Intel SSE2
SSE3	DEPRECATED. Intel® Streaming SIMD Extensions 3 (Intel® SSE3).
SSSE3	Supplemental Streaming SIMD Extensions 3 (SSSE3)
SSE4_1	Intel® Streaming SIMD Extensions 4-1 (Intel® SSE4-1)
SSE4_2	Intel® Streaming SIMD Extensions 4-2 (Intel® SSE4-2)
AVX	Intel® Advanced Vector Extensions (Intel® AVX)
AVX2	Intel® Advanced Vector Extensions 2 (Intel® AVX2)

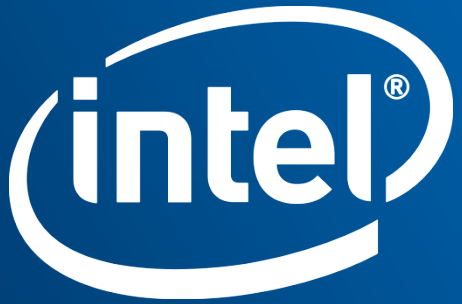
Intel® MKL Lab – BLAS Usage **Activity #4**

#Conditional Numerical Reproducibility - CNR

Run and record execution times

./3.out 8000

- export MKL_CBWR=AVX512 … Elapsed time = ….., sec
- export MKL_CBWR=AVX … Elapsed time = ……, sec
- export MKL_CBWR=SSE2 … Elapsed time = ……., sec
- export MKL_CBWR=COMPATIBLE …. Elapsed time = ……., sec


intel®

Legal Disclaimer & Optimization Notice

INFORMATION IN THIS DOCUMENT IS PROVIDED “AS IS”. NO LICENSE, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE, TO ANY INTELLECTUAL PROPERTY RIGHTS IS GRANTED BY THIS DOCUMENT. INTEL ASSUMES NO LIABILITY WHATSOEVER AND INTEL DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY, RELATING TO THIS INFORMATION INCLUDING LIABILITY OR WARRANTIES RELATING TO FITNESS FOR A PARTICULAR PURPOSE, MERCHANTABILITY, OR INFRINGEMENT OF ANY PATENT, COPYRIGHT OR OTHER INTELLECTUAL PROPERTY RIGHT.

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.

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