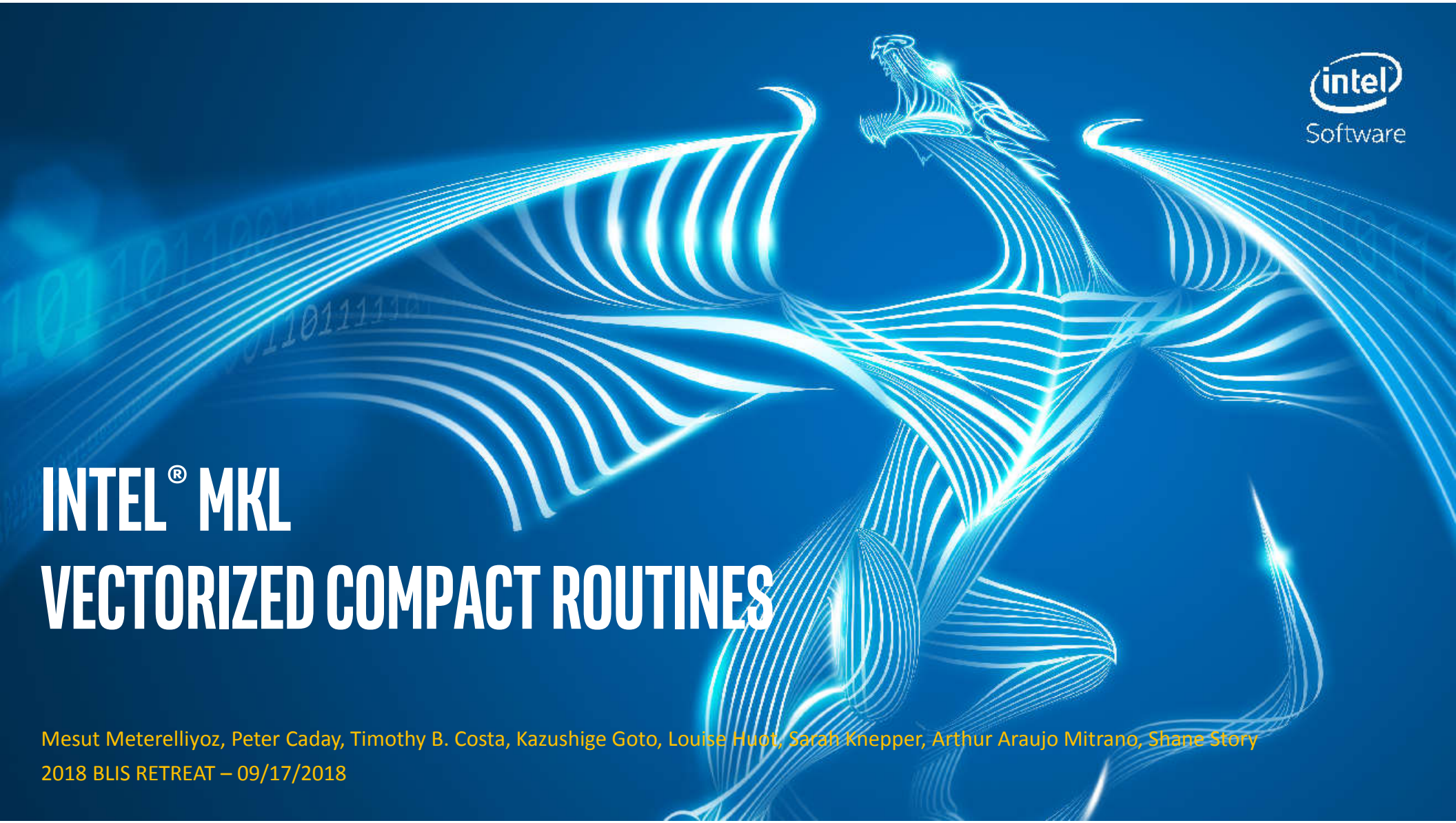


INTEL® MKL VECTORIZED COMPACT ROUTINES

Mesut Meterelliyoz, Peter Caday, Timothy B. Costa, Kazushige Goto, Louise Huot, Sarah Knepper, Arthur Araujo Mitrano, Shane Story

2018 BLIS RETREAT – 09/17/2018

OUTLINE

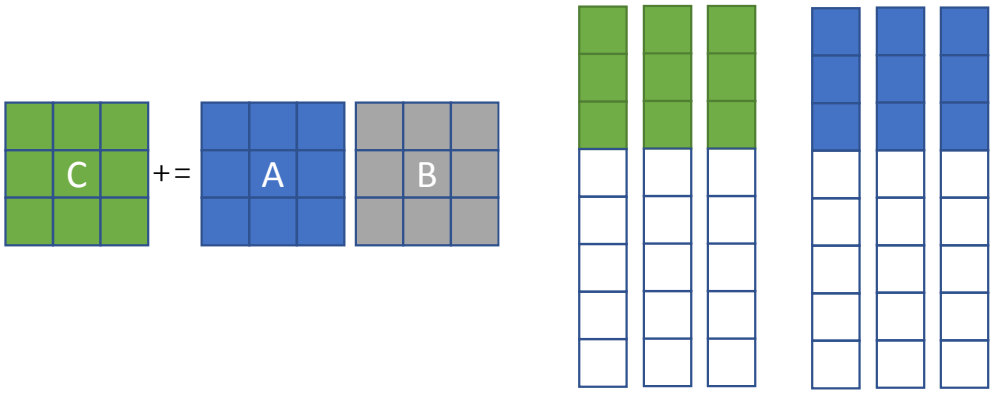
- Motivation
- Compact APIs
- Performance Results & Summary

CHALLENGES WITH SMALL MATRICES

- High function call and error checking overheads
- Limited vectorization opportunity and non-local data access for large leading dimensions

```
C = beta*C
DO i=1,(M/u)
   DO j=1,N
      DO kk=1,K
         C(i  ,j) += alpha*A(i,kk)*B(kk,j)
         C(i+1,j) += alpha*A(i+1,kk)*B(kk,j)
           .
           .
           .
         C(i+u,j) += alpha*A(i+u,kk)*B(kk,j)
      END DO
   END DO
END DO
```

3x3x3 DGEMM and Intel AVX512® register mapping



ZMM registers

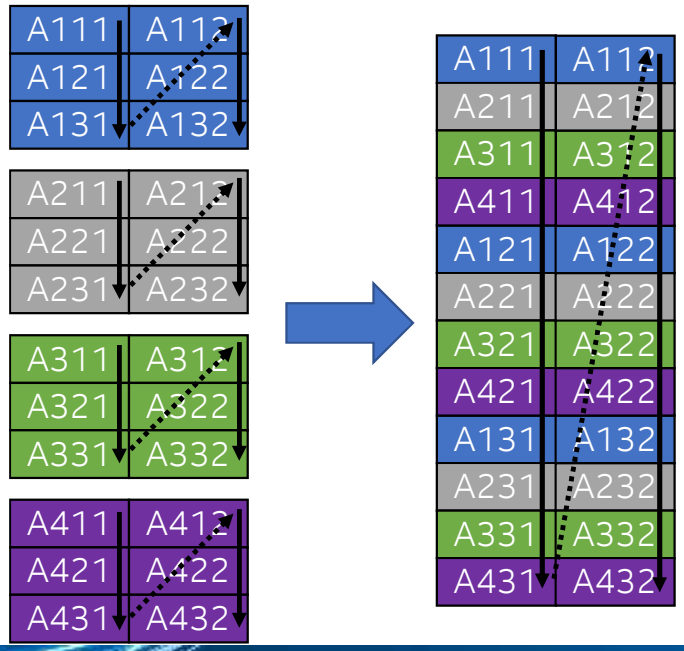
COMPACT API TO OVERCOME PERFORMANCE CHALLENGES

- Applications perform multiple BLAS/LAPACK operations on a large number of small matrices
  - Numerical factorization, blocked-sparse matrices, rotation matrices, finite element, and finite volume
- Challenges: limited vectorization, function call overheads, and error checking overheads
- Solution: Perform multiple BLAS/LAPACK operations using a new data layout (compact) amenable to vectorization
- Function call overheads and error checking is amortized over multiple BLAS/LAPACK operations
- Compact APIs
  - Functions to query the optimal format and memory required for the compact data layout
  - Matrix data layout transformation functions
  - Compute kernels: gemm, trsm, getrinp, getrfnp, potrf, geqrf

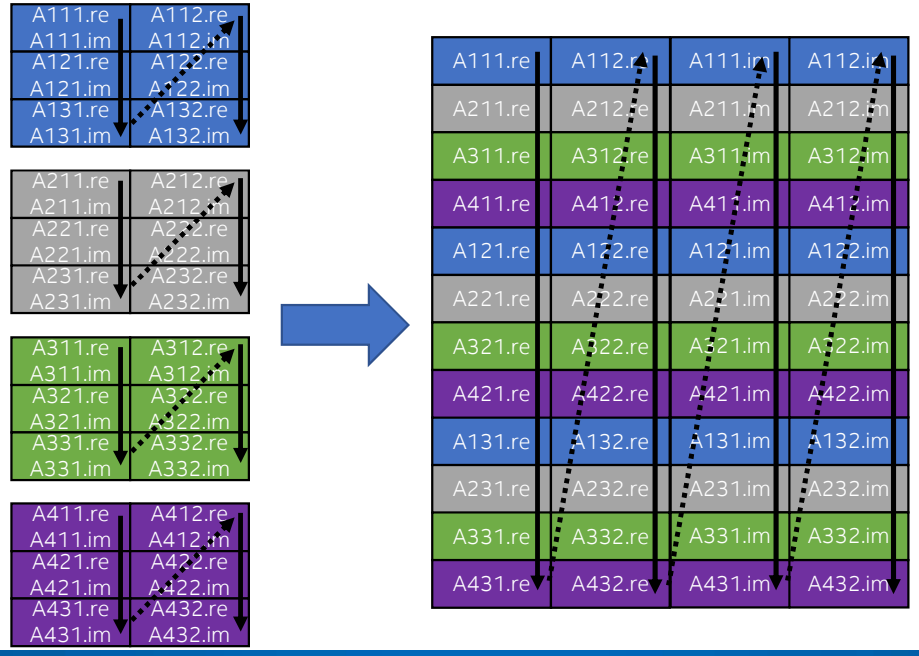
COMPACT DATA LAYOUT

- Matrix elements with same index are interleaved in memory
- Size of the subgroup is SIMD length to fully utilize SIMD instructions
- Example reformatting of 3x2 matrices with subgroup size = 4:

Real data type



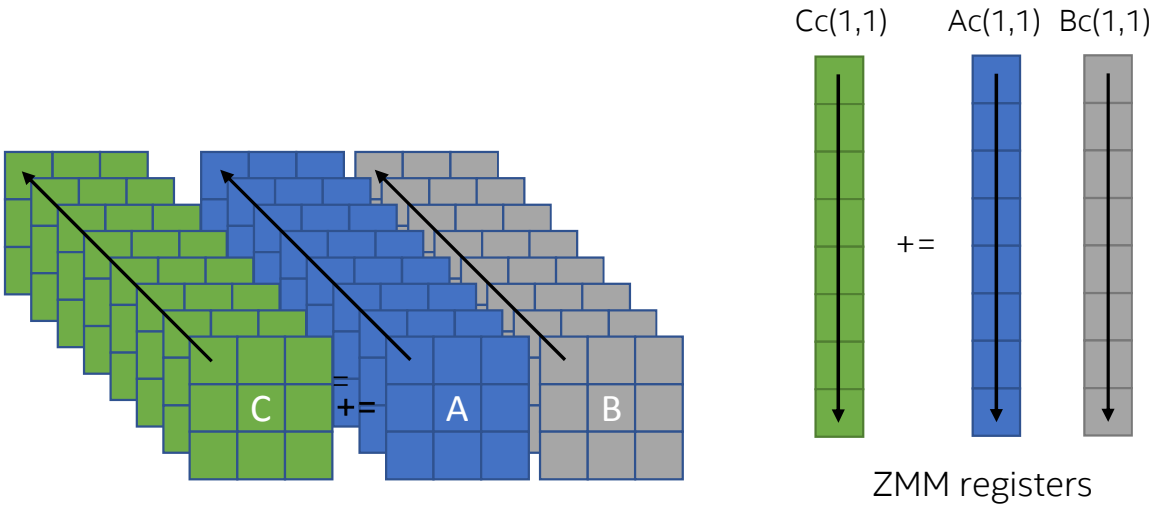
Complex data type



VECTORIZATION WITH COMPACT DATA LAYOUT

- Matrix elements with same col/row index loaded to a SIMD register
- Vectorization across the matrices becomes trivial
- Data padding if the number of matrices are not multiples of SIMD vector length

3x3x3 MKL_DGEMM_COMPACT and Intel AVX512 register mapping



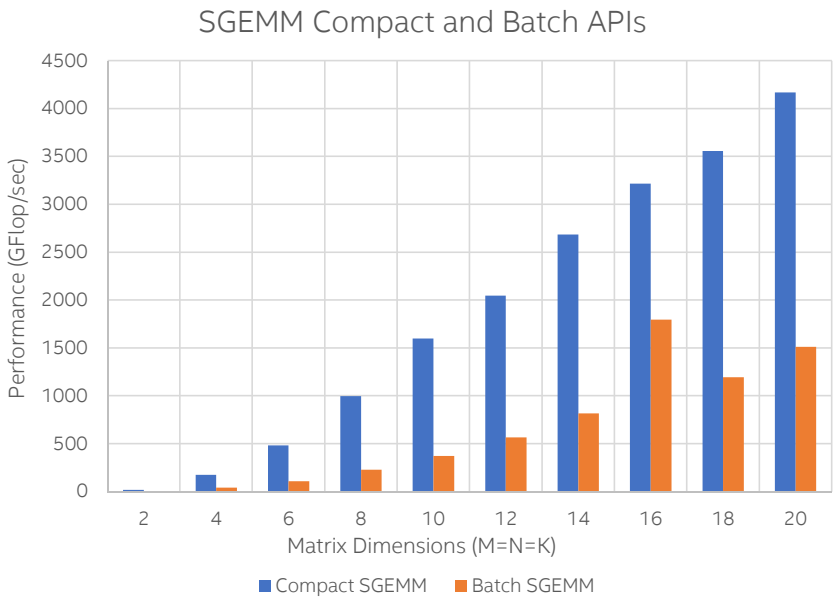
ZMM registers

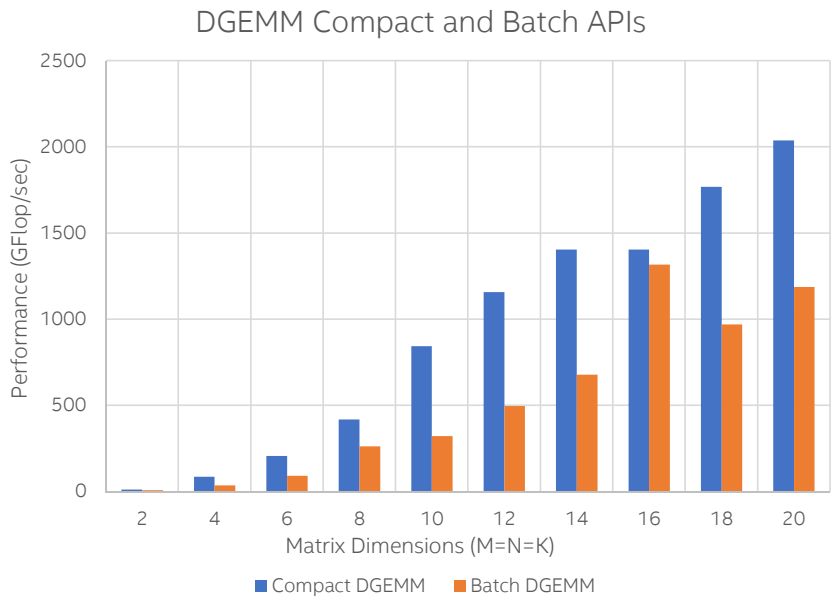
COMPACT API USAGE EXAMPLE

- Non-standard BLAS API that requires some code modification
- Intel MKL utility functions to transform matrices between column/row major and compact layout

```
#include <mkl.h>

// query the optimal format for the architecture
MKL_COMPACT_PACK compact_format = mkl_get_format_compact();

// query memory requirements and allocate memory for compact layout
a_size = mkl_dget_size_compact(lda, k, compact_format, num_matrix);
b_size = mkl_dget_size_compact(ldb, n, compact_format, num_matrix);
c_size = mkl_dget_size_compact(ldc, n, compact_format, num_matrix);

// transform the data into the compact format
mkl_dgepack_compact(layout, m, k, a_array, lda, a_c, lda, compact_format, num_matrix);
mkl_dgepack_compact(layout, k, n, b_array, ldb, b_c, ldb, compact_format, num_matrix);
mkl_dgepack_compact(layout, m, n, c_array, ldc, c_c, ldc, compact_format, num_matrix);

// computations on compact data layout
mkl_dtrsm_compact(layout, side, uplo, transa, diag, m, n, alpha, a_c, lda, b_c, ldb, compact_format, num_matrix);
mkl_dgemm_compact(layout, transa, transb, m, n, k, alpha, a_c, lda, b_c, ldb, beta, c_c, ldc, compact_format, num_matrix);

// transform from compact format to standard BLAS format
mkl_dgeunpack_compact(layout, m, n, c_array, ldc, c_c, ldc, compact_format, num_matrix);
```

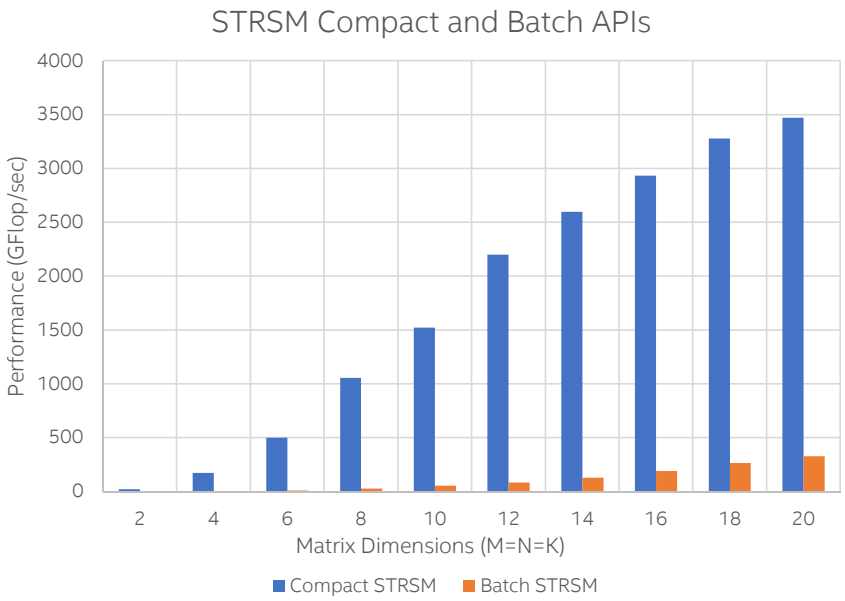
COMPACT API PERFORMANCE ON INTEL® XEON® PLATINUM PROCESSOR

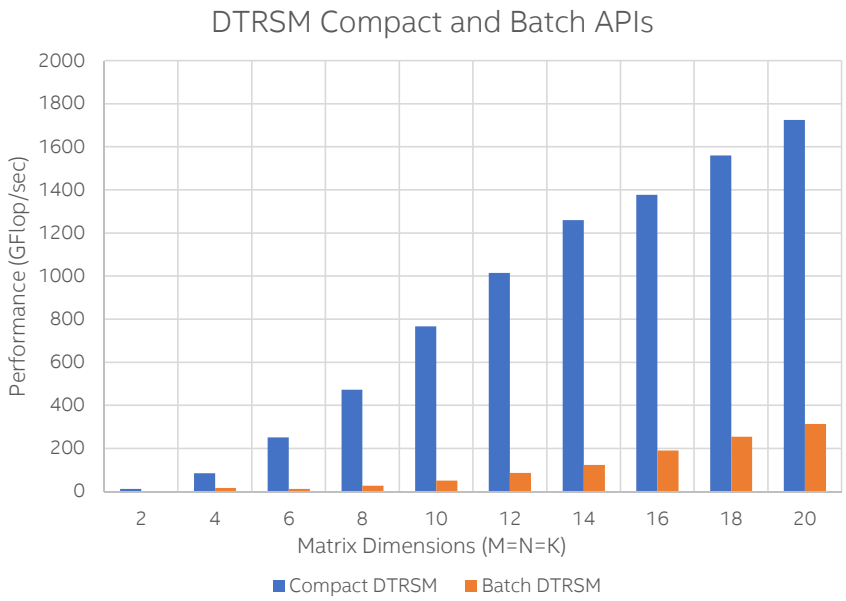




Configuration: Intel® Xeon® Platinum 8180, 2x28 cores, 2.5 GHz, 376 GB RAM, OS Ubuntu, 16.04 LTS; Intel® MKL 2018.
Performance results may not reflect all publicly available security updates. See configuration disclosure for details. No product can be absolutely secure. Software and workloads used in 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. For more complete information visit www.intel.com/benchmarks. Benchmark source: Intel® Corporation.
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

COMPACT API PERFORMANCE ON INTEL® XEON® PLATINUM PROCESSOR





Configuration: Intel® Xeon® Platinum 8180, 2x28 cores, 2.5 GHz, 376 GB RAM, OS Ubuntu, 16.04 LTS; Intel® MKL 2018.
Performance results may not reflect all publicly available security updates. See configuration disclosure for details. No product can be absolutely secure. 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. For more complete information visit www.intel.com/benchmarks. Benchmark source: Intel® Corporation.
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

SUMMARY

- Compact APIs are available starting from Intel MKL 2018
  - BLAS: gemm, trsm
  - LAPACK: getrinp, getrfnp, potrf, geqrf
- Perform enough computations to amortize transformation cost
- Intel MKL Developer Reference for more details and other small matrix solutions

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.

Performance results may not reflect all publicly available security updates. See configuration disclosure for details. No product can be absolutely secure. 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


intel
Software