Overview of Intel® MKL Sparse BLAS

Software and Services Group
Intel Corporation
Agenda

- Why and when sparse routines should be used instead of dense ones?
- Intel MKL Sparse BLAS functionality
- Sparse Matrix Storage formats
- Naming conventions in Sparse BLAS
- Impact of hardware on the performance of sparse computations
- Performance charts
- Summary
Why and when sparse routines should be used instead of dense ones?

There are 2 main reasons why Sparse BLAS can be better than dense BLAS

1. Sparse matrix usually requires significantly less memory for storing data.
2. Sparse routines usually spend less time to compute result for sparse data.

Sparse BLAS performance often seems not as high as performance of the dense one. For example, BLAS level 2 performance on modern CPU’s like Intel® Xeon® E5-2680 could be measured in dozens of Gflop/s, while Sparse BLAS level 2 performance typically equals to a few Gflop/s. Let’s consider the computational cost of matrix-vector multiply $y=y+A*x$.

If the matrix is represented in a sparse format, the computational cost is $2*\text{nnz}$, where nnz is the number of non-zero elements in matrix A. If the matrix is represented in a dense format, the computational cost is $2*n*n$. Let $\text{nnz}$ be $9*n$. Then $(\text{computational cost for dense matrix-vector multiply}) / (\text{computational cost sparse matrix-vector multiply})$ is $n/9$. Therefore, a direct comparison of sparse and dense BLAS performance in Gflop/s can be misleading as total time spend in Sparse BLAS computations can be less than the total time spent in dense BLAS despite of higher GFlop/s for dense BLAS functionality.
Intel MKL Sparse BLAS functionality

1. Sparse matrix-vector/matrix-matrix product:
   \[ C \leftarrow \alpha \cdot \text{op}(A) \cdot B + \beta \cdot C \quad \text{or} \quad C \leftarrow \text{op}(A) \cdot B \quad \text{(Level 2 only)} \]

2. Solving a triangular system
   \[ C \leftarrow \alpha \cdot \text{inverse of (op(A))} \cdot B \quad \text{or} \quad C \leftarrow \text{inverse of (op(A))} \cdot B \quad \text{(Level 2 only)} \]

Where, C and B are dense matrices or vectors, A is a sparse matrix, alpha and beta are scalars, op(A) is one of the possible operations:

   \[ \text{op}(A) = A \quad \text{or} \quad \text{op}(A) = \text{(conjugate) transpose of } A. \]

3. Auxiliary routines: Sparse BLAS converters, Level 1 routines, addition and multiplication of two sparse matrices.

Intel MKL Sparse BLAS supports all types of precision, 0-based (C-style) and 1-based (FORTRAN-style) indexing.
Sparse Matrix Storage formats

Intel® MKL supports 6 sparse matrix storage format (CSR, CSC, COO, DIA, SKY, BSR):

Point-entry formats:

CSR: Compressed Sparse Row format
CSC: Compressed Sparse Column format
COO: Coordinate format
DIA: Diagonal format
SKY: Skyline format. Used for storing triangular factors in various LU decompositions.

Block-entry format:

BSR: Block sparse row format. This format is similar to the CSR format. Nonzero entries in the BSR are square dense blocks.
Naming conventions in Sparse BLAS

The routines with typical (conventional) interface have six-character base names in accordance with the template:

\[ \text{mkl}_{<\text{precision}>_{<\text{data}>_{<\text{operation}>}(\ )},} \]

where \(<\text{precision}>\) field denotes type of precision (\(c, d, s\) or \(z\)), \(<\text{data}>\) is the sparse matrix storage format, the \(<\text{operation}>\) field indicates the type of operation (\(\text{mv}\) is for matrix-vector multiply, \(\text{mm}\) is for matrix-matrix, \(\text{sv}\) is for triangular solver, \(\text{sm}\) is for triangular solver for many right hand sides). These type routines can perform a given operation for 6 types of matrices (general, symmetric, triangular, skew-symmetric, Hermitian and diagonal).
Example

$mkl_dcsrmv(\text{transa}, m, k, \alpha, \text{matdescra}, \text{val}, \text{indx}, \text{pntrb}, \text{pntre}, x, \beta, y)$

is

- double precision ($mkl_dcsrmv$)
- compressed sparse row storage ($mkl_dcsrmv$)
- matrix-vector multiply ($mkl_dcsrmv$)

Array $\text{matdescra}$ describes the relevant characteristics of a input matrix: matrix type, upper/lower triangular indicator, main diagonal type (unit/non-unit) and type of indexing.
Naming conventions in Sparse BLAS (part II)

The routines with simplified interfaces have eight-character base names in accordance with the templates:

\[ mkl_<\text{precision}> <data> <mtype> <operation>( ) \]

for routines with one-based indexing; and

\[ mkl_cspblas_<\text{precision}> <data> <mtype> <operation>( ) \]

for routines with zero-based indexing.

Possible values for the field \(<mtype>\)

- **ge** - sparse matrix-vector for general matrix
- **sy** - sparse matrix-vector for symmetric matrix
- **tr** - triangular solver
Impact of hardware on the performance of sparse computations.

- Operations on data in cache are cheap.
- OpenMP parallelization on the latest Intel® architectures becomes highly efficient even for Level 1-2 BLAS.
- Transferring data from RAM to cache is expensive.
- Contiguous memory patterns in a sparse representation provide an opportunity to use SSE/AVX vector instructions efficiently.
- The memory bandwidth is usually a performance bottleneck for Sparse BLAS. Hence, codes should try to use Sparse BLAS formats with contiguous memory patterns (for example BSR, SKY, or DIA) to simplify vectorization & reuse data in cache as opposed to Sparse formats with possibly non-contiguous memory pattern like CSR or CSC.
Performance tests and ratings are measured using specific computer systems and/or components and reflect the approximate performance of Intel products as measured by those tests. Any difference in system hardware or software design or configuration may affect actual performance. Buyers should consult other sources of information to evaluate the performance of systems or components they are considering purchasing. For more information on performance tests and on the performance of Intel products, refer to [http://www.intel.com/content/www/us/en/benchmarks/resources-benchmark-limitations.html](http://www.intel.com/content/www/us/en/benchmarks/resources-benchmark-limitations.html)


*Other brands and names are the property of their respective owners.
Performance tests and ratings are measured using specific computer systems and/or components and reflect the approximate performance of Intel products as measured by those tests. Any difference in system hardware or software design or configuration may affect actual performance. Buyers should consult other sources of information to evaluate the performance of systems or components they are considering purchasing. For more information on performance tests and on the performance of Intel products, refer to http://www.intel.com/content/www/us/en/benchmarks/resources-benchmark-limitations.html

Refer to our Optimization Notice for more information regarding performance and optimization choices in Intel software products at: http://software.intel.com/en-us/articles/optimization-notice/

*Other brands and names are the property of their respective owners.
References

Intel® MKL product Information
- [www.intel.com/software/products/mkl](http://www.intel.com/software/products/mkl)

Technical Issues/Questions/Feedback
- [http://premier.intel.com/](http://premier.intel.com/)

Self-help
  (Click “Support Resources” tab)

User Discussion Forum
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.

Copyright ©, Intel Corporation. All rights reserved. Intel, the Intel logo, Xeon, Core, VTune, and Cilk
are trademarks of Intel Corporation in the U.S. and other countries.

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