Intel® Math Kernel Library 7.2.1 for Linux Release Notes*

Contents

Overview

The Intel® Math Kernel Library (Intel® MKL) provides developers of scientific, engineering and financial software with a set of linear algebra routines, discrete Fourier transforms and vectorized math and random number generation functions, all optimized for the latest Intel® Pentium® 4 processors, Intel® Xeon™ processors with Streaming SIMD Extensions 3 (SSE3) and Intel® Extended Memory 64 Technology (Intel® EM64T), and Intel® Itanium® 2 processors. Intel MKL provides linear algebra functionality with LAPACK (solvers and eigensolvers) plus levels 1, 2, and 3 BLAS offering the vector, vector-matrix, and matrix-matrix operations needed for complex mathematical software. For solving sparse systems of equations, Intel MKL now provides a direct sparse solver, for which two interfaces are provided: the PARDISO interface and the DSS interface. Intel MKL offers multidimensional discrete Fourier transforms (1D, 2D, 3D) with mixed radix support (not limited to sizes of powers of 2). Intel MKL also includes a set of vectorized transcendental functions (called the Vector Math Library (VML)) offering both greater performance and excellent accuracy compared to the libm (scalar) functions for most of the processors. The Vector Statistical Library (VSL) offers high performance, hand tuned vectorized random number generators for a number of probability distributions. Intel MKL offers multi-threading support using OpenMP* in addition to being a fully thread-safe library.

Performance on Intel® processors has improved on a number of functions, as noted below. In addition, this version of MKL also performs superbly on non-Intel (X86) processors allowing software developers to use a single library regardless of what hardware the customer uses.

Version 7.2.1 of Intel MKL introduces:

- Performance improvements and bug fixes.

For detailed information on these features, please refer to the "New in Intel® MKL 7.2.1" section below.

The original versions of the BLAS from which that part of Intel MKL was derived can be obtained from http://www.netlib.org/blas/index.html. The original versions of LAPACK from which that part of Intel MKL was derived can be obtained from http://www.netlib.org/lapack/index.html. The authors of LAPACK are E. Anderson, Z. Bai, C. Bischof, S. Blackford, J. Demmel, J. Dongarra, J. Du Croz, A. Greenbaum, S. Hammarling, A. McKenney, and D. Sorensen.

Directory Structure

The information below indicates the high level structure for Intel MKL.

mkl721	Main directory
mklnotes.htm	Release notes (this file)
mkllic.htm	Intel MKL license
redist.txt	List of redistributable files
mkl721/doc	Directory for documents
index.htm	Index to the Intel MKL documentation
mklman.pdf	Intel MKL manual
mkluse.htm	User notes for Intel MKL
vmlnotes.htm	General discussion of VML
vslnotes.pdf	General discussion of VSL
mkl721/examples	Source and data for examples
mkl721/include	Contains include files for both library routines and test and example programs
mkl721/tests	Source and data for tests
mkl721/lib/32	Contains static libraries and shared objects for IA-32 applications
mkl721/lib/em64t	Contains static libraries and shared objects for applications running on processors with Intel EM64T
mkl721/lib/64	Contains static libraries and shared objects for the Itanium® 2 processor
mkl721/tools/builder	Contains tools for creating custom dynamically linkable libraries

mkl721/tools/environment	Contains shell scripts to set environment variables in the user shell
mkl721/tools/support	Contains a utility for reporting package ID and license key information to Intel® Premier Support

New in Intel® MKL 7.2.1

- Performance improvements since Intel® MKL 7.2
  - Improvements for the Intel® Itanium® 2 processor
    - BLAS
      - DGEMM: 1-3% improvement for TN and TT cases, and increased consistency of multi-threaded performance
      - *TRMV: 20-300% improvement
      - ZGERC, ZGERU: 20-30% improvement
    - VML
      - vdPowx: improved by 19 times for arguments 0^b (b constant).
  - Improvements for the Intel® Pentium® 4 processor and Intel® Xeon™ processor with SSE3.
    - BLAS
      - DGEMM: 3-10% improvement
      - CGEMM: 35-50% improvement
      - ZGEMM: 35-50% improvement
      - DTRSM: 10-60% improvement
      - DSYMM: 3-10% improvement
      - SDOT, CDOT(U,C): 200% improvement for problems fitting in L1 cache, 30-100% in L2 cache, and 20-40% in memory
      - SASUM: 500% faster
      - (C,CS)SCAL, (Z,ZD)SCAL: 15-25% faster
      - SCASUM: 6% faster
      - CGEMV: more than 200% faster
      - ZGEMV: 26% faster
      - (C,Z)HEMV: 15-60% faster
    - DFT
      - 1D DFT with double precision complex data: 8-14% faster
  - LAPACK Optimizations
    - DGESVD: 200-1000% improvement
    - *EVD, *ORGQR: Threaded
    - Improved the accuracy of single precision routines for linear least squares problems

- Other improvements
  - Intel MKL is in the process of unifying how NaN’s are handled within the BLAS. This work is not complete so NaN responses may be different in version 7.2.1 than in previous versions.

- The memory manager has been repaired so that level 3 BLAS will work correctly on large NUMA systems. There is a limit on the maximum number of threads that can be created in the current version of Intel MKL. This will be resolved in a future version of Intel MKL. Contact customer support if this is a problem for your application.
- Built direct sparse solver (PARDISO) with -fpic flag for EM64T
- Intel MKL now supports the "medium" addressing model for its static archives on 64-bit Intel® Xeon processors (i.e., compiled with "-mcmodel=medium" flag

System Requirements

Recommended hardware: a PC, workstation or server, with Intel® Xeon™ processor, Intel® Xeon™ processors with SSE3 and Intel EM64T, Pentium 4 processor, or Itanium® 2 processor.

Software requirements

A supported OS (Intel MKL has been tested with the following):

- Red Hat* Linux* version 9.0 (on IA-32 systems only)
- Red Hat* EL 3.0
- SuSE* Linux* 8.2 (on IA-32 systems only)
- SuSE* Linux* Enterprise Server 8
- Intel EM64T: Red Hat Enterprise Linux 3.0 AS
- Intel EM64T: Red Hat Enterprise Linux 4.0 AS
- Intel EM64T: SuSE* Linux* Enterprise Server 9

A supported C or Fortran compiler (Intel MKL has been tested with the following):

- Intel® C++ Compiler versions 8.0 and 8.1 for IA-32 and Itanium® processors
- Intel® Fortran Compiler versions 8.0 and 8.1 for IA-32 and Itanium® processors
- Intel® C++ Compiler version 8.1 Extended Memory 64 Technology Edition
- Intel® Fortran Compiler version 8.1 Extended Memory 64 Technology Edition
- GNU compiler collection

Note: Intel Cluster MKL has parts which have Fortran interfaces, and are Fortran in their data structures, and parts which have C interfaces and have C data structures. The user notes file (mkluse.htm in the doc directory) contains advice on how to link to Intel Cluster MKL with different compilers.

Installation

To install the Intel MKL package on Linux*, use the following instructions. The installation

software installs the full Intel MKL file set for all supported processors. See the Intel MKL website for updates, when available.

1. Use the tar command to extract the Intel MKL package in a directory to which you have write access
   (e.g., `tar -zxvf` *package.tar.gz*).
2. Become the root user and execute the install script in the directory where the tar file was extracted by typing "`./install`".
   - The use of RPM necessitates root access to your system. If you do not have root access, you can find the extracted RPM package in a temporary directory. You may also contact customer support for work around information.
   - If you are installing an evaluation version of Intel MKL you may have to take a few steps to get the FLEXlm key license set up on your system. Instructions are provided by the installer.
3. The Intel® Performance Libraries products already installed will be listed, followed by a menu of products to install which includes:
   - Intel® Math Kernel Library Version 7.2.1
4. Select a package to install. Before the installation begins, you will be asked to enter a temporary directory in which the contents of the self-extracting installation file will be placed before actual installation begins. After installation, the files will still be located in this temp directory and must be deleted manually. All packages needed to use the product will also be installed. The recommended (default) installation directory is `/opt/intel`. In the directory you choose, a directory named `mkl721` will be created and all files will be installed there. Any previous version of Intel MKL may remain installed when installing Intel MKL 7.2.1, but you will be required to remove the beta version of this software if you have it installed. Be sure to update your build scripts to point to the desired version of Intel MKL if you choose to keep multiple versions installed.
   - The Intel MKL installation program uses RPM as the installation vehicle. Some versions of RPM do not allow redirection of installation. If the install program detects that you have a version of RPM that does not allow redirection, you will be required to install to the default directory.
5. After installation, the packages installed will be redisplayed, followed by a redisplay of the install menu. Enter 'x' to exit the install script.

Three files, mklvars32.sh, mklvarsem64t.sh, and mklvars64.sh, will be placed in the tools/environment directory. These files can be used to set the INCLUDE and LD_LIBRARY_PATH environment variables in the current user shell.

Intel MKL uses Macrovision's* FLEXlm* electronic licensing technology. License management should be transparent, but if you have any problems during installation, please make sure a current license file (*.lic) is located in the same directory as the **install** file. If you

still have problems, please submit an issue to Intel® Premier Support. See the "Technical Support and Feedback" section of this document for details.

Known Limitations

Limitations to the sparse solver in Intel MKL 7.2.1:

1. The default number of threads (when OMP_NUM_THREADS is not set) is equal to the number of processors in the system. This differs from the default OpenMP mode in Intel MKL (by default the number of threads is set to one).
2. Only statically linkable sparse solver library files will be available with this release.
3. Enhanced precision accumulation is implemented in long doubles (10 bytes real precision).
4. Statistics output is not implemented (msglvl=1 will not deliver statistics).

There are a number of limitations in the current implementation of the set of DFT functions:

1. The function DftiCopyDescriptor is not implemented.
2. The function DftiGetValue is implemented with the following restriction: the DFTI_FORWARD_ORDERING and DFTI_BACKWARD_ORDERING parameters are not yet supported.
3. Complex data is stored using the Fortran data type; real and imaginary parts are adjacent.
4. Modes DFTI_INITIALIZATION_EFFORT, DFTI_WORKSPACE, and DFTI_TRANSPOSE are implemented only for the default case. DFTI_FORWARD_SIGN can have the default value only and is not changeable by the DftiSetValue function.
5. DFTI_PRECISION, DFTI_DIMENSION, and DFTI_LENGTHS are settable only through the DftiCreateDescriptor function and are not changeable by the DftiSetValue function.
6. Mode DFTI_FORWARD_DOMAIN can not have the value DFTI_CONJUGATE_EVEN.
7. 3D real DFT is not currently implemented.
8. Modes DFTI_REAL_STORAGE and DFTI_CONJUGATE_EVEN_STORAGE can have the default value only and are not changeable by the DftiSetValue function (i.e., DFTI_REAL_STORAGE = DFTI_REAL_REAL and DFTI_CONJUGATE_EVEN_STORAGE = DFTI_COMPLEX_REAL).
9. Mode DFTI_COMPLEX_STORAGE can have the default value only and is not changeable by the DftiSetValue function. In other words, DFTI_COMPLEX_STORAGE is always DFTI_COMPLEX_COMPLEX.

When using the DFTs in Intel MKL it may be necessary to explicitly link 'libm'. Please include '-lm' on your link line after any reference to Intel MKL library files.

On Intel® processors with Intel® EM64T enabled, user programs compiled with the GNU Fortran compiler (version 3.2.3) will likely get incorrect results from those BLAS level 1 functions in Intel® MKL that return single precision values. The GNU Fortran compiler expects REAL*4 values in the first 8 bytes of the return register (just as a double precision value would be represented) while the Intel® Fortran compiler expects REAL*4 values in the first 4 bytes of the return register. The behavior of Intel MKL is compatible with that of the Intel Fortran compiler.

Hyper-Threading Technology (HT Technology) is especially effective when each thread is performing different types of operations and when there are under-utilized resources on the processor. Intel MKL fits neither of these criteria as the threaded portions of the library execute at high efficiencies (using most of the available resources) and perform identical operations on each thread. You may obtain higher performance when using Intel MKL without HT Technology enabled.

DFT, VML, and VSL functions can not be used with Fortran 77 compilers.

Memory Allocation: In order to achieve better performance, memory allocated by Intel MKL is not released. This behavior is by design and is a one time occurrence for Intel MKL routines that require workspace memory buffers. Even so, the user should be aware that some tools may report this as a memory leak. Should the user wish, memory can be released by the user program through use of a function (`MKL_FreeBuffers()`) made available in Intel MKL or memory can be released after each call by setting an environment variable (`MKL_DISABLE_FAST_MM`) (see technical user notes in the `doc` directory for more details). Using one of these methods to release memory will not necessarily stop programs from reporting memory leaks, and in fact may increase the number of such reports should you make multiple calls to the library thereby requiring new allocations with each call. Memory not released by one of the methods described will be released by the system when the program ends. The maximum number of buffers allocated in each thread is 32. To avoid this restriction disable memory management as described above.

On Red Hat* Enterprise Linux 3.0, in order to ensure that the correct support libraries are linked, the environment variable LD_ASSUME_KERNEL must be set: For example: 'export LD_ASSUME_KERNEL=2.4.1'

Technical Support and Feedback

Self Help and User Forums

A rich repository of self-help product information such as tutorials, getting started tips, known product issues, product errata, compatibility information and answers to frequently asked questions can be found at the Intel® Software Development Products Technical Support. It's a great place to find answers quickly or to gain insight in using our products effectively.

The Intel MKL User Forum is the place to ask questions of and share information with other users of Intel® MKL.

Submitting Issues

Your feedback is very important to us. To receive technical support and product updates for the tools provided in this product you need to register at the Intel® Registration Center and click on "Create New Account".

For information about the Intel® MKL including FAQ's, tips and tricks, and other support information, please visit: http://support.intel.com/support/performancetools/libraries/mkl

Note: If you are having trouble registering or unable to access your Premier Support account, contact developer.support@intel.com. Please do not email your technical issue to developer.support@intel.com as it is not a secure medium.

To submit an issue via the Intel® Premier Support website, please perform the following steps:

1. Ensure that Java* and JavaScript* are enabled in your browser.
2. Go to https://premier.intel.com/.
3. Type in your Login and Password. Both are case-sensitive.
4. Click the "Submit Issues" button.
5. Read the Confidentiality Statement and click the "I Accept" button.
6. Click on the "Go" button next to the "Product" drop-down list.
7. Click on the "Submit Issue" link in the left navigation bar.
8. Choose "Development Environment (tools,SDV,EAP)" from the "Product Type" drop-down list.
9. If this is a software or license-related issue choose "**Intel(R) MKL for Linux***" from the "Product Name" drop-down list.
10. Enter your question and complete the fields in the windows that follow to successfully submit the issue.

Please follow these guidelines when forming your problem report or product suggestion:

1. Describe your difficulty or suggestion.

For problem reports please be as specific as possible (e.g., including compiler and link command line options), so that we may reproduce the problem. Please include a small test case if possible.

2. Describe your system configuration information.
   Be sure to include specific information that may be applicable to your setup: operating system, name and version number of installed applications, and anything else that may be relevant to helping us address your concern.

Related Products and Services

Information on Intel® software development products is available at http://www.intel.com/software/products. Some of the related products include:

- The Intel® Software College provides interactive tutorials, documentation, and code samples that teach Intel® architecture and software optimization techniques.
- The VTune™ Performance Analyzer allows you to evaluate how your application is utilizing the CPU and helps you determine if there are modifications you can make to improve your application's performance.
- The Intel® C++ and Fortran Compilers are an important part of making software run at top speeds and fully support the latest Intel IA-32 and Itanium® processors.
- The Intel® Performance Library Suite provides a set of routines optimized for various Intel® processors. The Intel® Math Kernel Library, which provides developers of scientific and engineering software with a set of linear algebra, fast Fourier transforms and vector math functions optimized for the latest Intel Pentium and Intel Itanium® processors. The Intel® Integrated Performance Primitives consists of cross platform tools to build high performance software for several Intel architectures and several operating systems.

Disclaimer and Legal Information

DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY, RELATING TO SALE AND/OR USE OF INTEL PRODUCTS INCLUDING LIABILITY OR WARRANTIES RELATING TO FITNESS FOR A PARTICULAR PURPOSE, MERCHANTABILITY, OR INFRINGEMENT OF ANY PATENT, COPYRIGHT OR OTHER INTELLECTUAL PROPERTY RIGHT. Intel products are not intended for use in medical, life saving, life sustaining, critical control or safety systems, or in nuclear facility applications.

Designers must not rely on the absence or characteristics of any features or instructions marked "reserved" or "undefined." Intel reserves these for future definition and shall have no responsibility whatsoever for conflicts or incompatibilities arising from future changes to them.

The software described in this document may contain software defects which may cause the product to deviate from published specifications. Current characterized software defects are available on request.

Intel, the Intel logo, Intel SpeedStep, Intel NetBurst, Intel NetStructure, MMX, i386, i486, Intel386, Intel486, Intel740, IntelDX2, IntelDX4, IntelSX2,Celeron, Intel Centrino, Intel Xeon, Intel XScale, Itanium, Pentium, Pentium II Xeon, Pentium III Xeon, Pentium M, and VTune are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

* Other names and brands may be claimed as the property of others.