Intel Math Kernel Library (MKL)

The Intel Math Kernel Library (MKL) is composed of highly optimized mathematical functions for engineering and scientific applications requiring high performance on Intel platforms. The functional areas of the library include linear algebra consisting of LAPACK and BLAS, fast Fourier transform (FFT), deep neural networks, vector statistics and data fitting, vector math, and miscellaneous solvers.

MKL is part of the Intel compiler releases. Once you load a compiler module (for example, **`module load comp-intel`**) the environment variable MKLROOT is automatically set and the path to the MKL library is automatically included in your default path.

A Layered Model for MKL

Intel employs a layered model for MKL. There are three layers: the interface layer, the threading layer, and the computational layer.

Interface Layer

The LP64 interface is for using 32-bit integer type and the ILP64 interface is for using 64-bit integer type.

Note: The SP2DP interface (used for mapping single-precision in Cray-style naming application and double-precision in MKL) has not been included in the MKL library releases since MKL 2017.

Threading Layers

Sequential

The sequential (non-threaded) mode does not require an OpenMP runtime library, and does not respond to the environment variable OMP_NUM_THREADS or its Intel MKL equivalents. In this mode, Intel MKL runs unthreaded code. However, it is thread-safe, which means that you can use it in a parallel region from your own OpenMP code.

You should use the library in the sequential mode only if you have a particular reason not to use Intel MKL threading. The sequential mode may be helpful when using Intel MKL with programs threaded with some non-Intel compilers or in other situations where you might need a non-threaded version of the library (for instance, in some MPI cases).

Note: The "sequential" library depends on the POSIX threads library (**`pthread`**), which is used to make the Intel MKL software thread-safe and should be listed on the link line.

Threaded

The "threaded" library supports the implementation of OpenMP and Threading Building Blocks (TBB), which Intel and GNU compilers provide. With the PGI compilers, only the OpenMP threading is available.

Computational Layer

For any given processor architecture (IA-32 or Intel 64) and operating system, this layer has only one computational library to link with, regardless of the interface and threading layer.

Compiler Support Runtime Libraries

Dynamically link the libiomp5 or libtbb library even if you link other libraries statically. Linking to the libiomp5 statically can be problematic, because the more complex your operating environment or application, the more likely redundant copies of the library are included. This may result in performance issues (oversubscription of threads) and even incorrect results. To link libiomp5 or libtbb dynamically, be sure the LD_LIBRARY_PATH environment variable is defined correctly.

Compiling and Linking with Intel MKL

Use the online Intel MKL Link Line Advisor to determine the libraries and options to specify on your link or compilation line.

For example, to do a dynamic linking and use parallel Intel MKL supporting LP64 interface for an OpenMP code, the MKL Link Line Advisor suggests:

```
-L$/lib/intel64 -lmkl_intel_lp64 -lmkl_intel_thread -lmkl_core -liomp5 -lpthread -lm -ldl
```

The -mkl Switch

The Intel compiler provides an **-mkl** switch to link to certain parts of the MKL library. In many cases, using this switch (instead of having to explicitly specify the MKL libraries) is all you need.

```
-mkl[=]
          link to the Intel(R) Math Kernel Library (Intel(R) MKL) and
          bring in the associated headers
            parallel   - link using the threaded Intel(R) MKL libraries.
                         This is the default when -mkl is specified
            sequential - link using the non-threaded Intel(R) MKL libraries
            cluster    - link using the Intel(R) MKL Cluster libraries plus
                         the sequential Intel(R) MKL libraries
```

The libraries that are linked in for:

```
    * -mkl=parallel

          --start-group \
          -lmkl_intel_lp64 \
          -lmkl_intel_thread \
          -lmkl_core \
          -liomp5 \
          --end-group \

    * -mkl=sequential

          --start-group \
          -lmkl_intel_lp64 \
          -lmkl_sequential \
          -lmkl_core \
          --end-group \

    * -mkl=cluster
```

```
--start-group \
-lmkl_intel_lp64 \
-lmkl_cdft_core \
-lmkl_scalapack_lp64 \
-lmkl_blacs_lp64 \
-lmkl_sequential \
-lmkl_core \
-liomp5 \
--end-group \
```

Note: Using **`--start-group`** and **`--end-group`** allows cycling through the enclosed libraries until all their inter-references (if any) have been resolved.

Where to find more information about MKL

The Intel Math Kernel Library Documentation includes HTML and PDF versions of the following guides:

- *Developer References for C and Fortran* - detailed function descriptions, including calling syntax
- *Developer Guides* - MKL usage information in greater detail

---

Article ID: 90
Last updated: 19 Jun, 2018
Revision: 25
Porting/Building Code -> Math & Scientific Libraries -> Intel Math Kernel Library (MKL)
https://www.nas.nasa.gov/hecc/support/kb/entry/90/