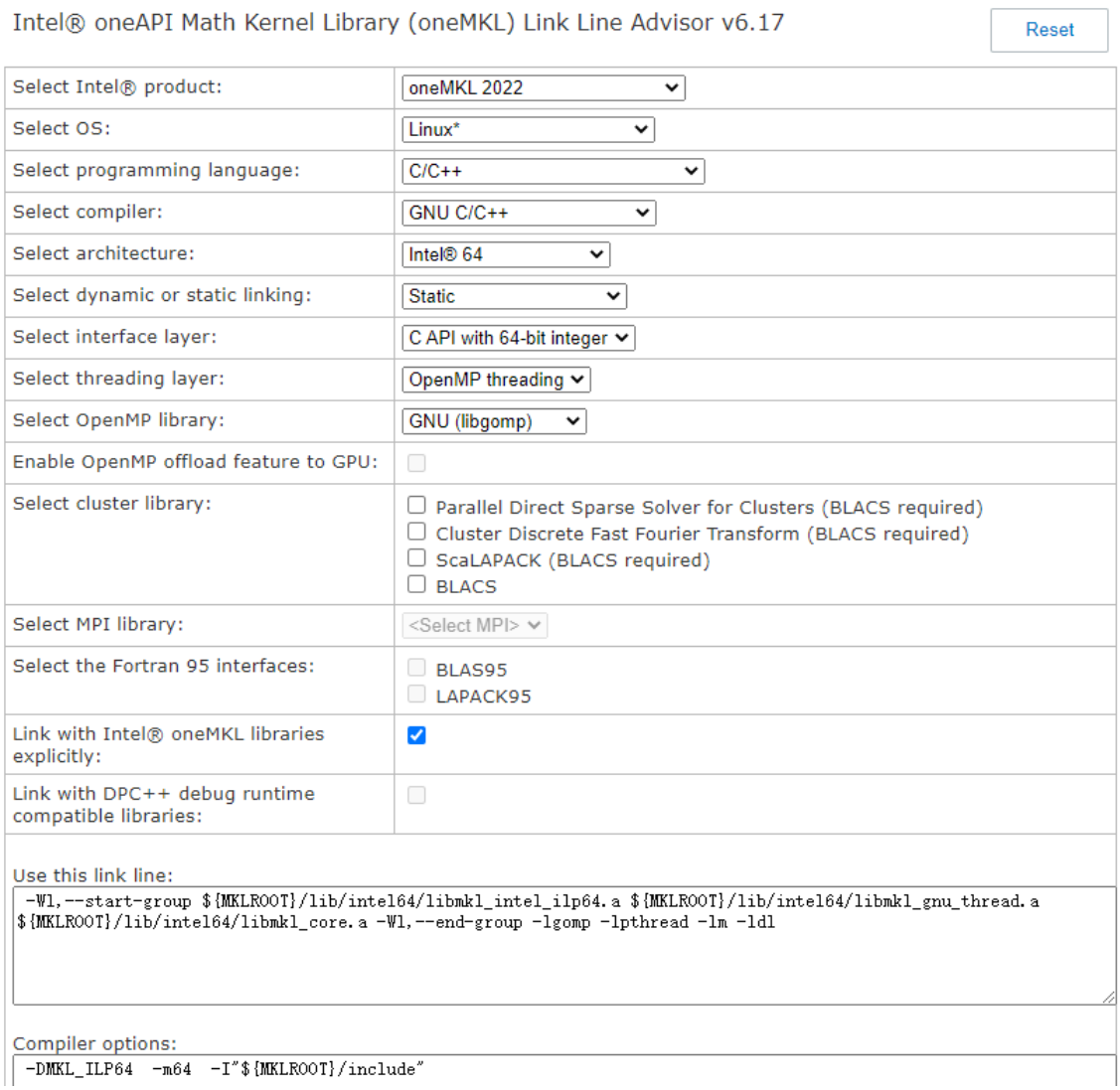
在使用MKL时，可以根据Intel® oneAPI Math Kernel Library Link Line Advisor的来确定编译选项，如下图所示：



但是在实际用这个编译选项时，却有些问题，无法跑多线程，以下是测试代码：

```
#include <stdio.h>
#include <omp.h>
#include "mkl.h"

#define M 384
#define N 384
#define K 384

void report_num_threads(int level)
{
    #pragma omp parallel num_threads(2)
    {
        //2 sub threads for each omp level
        printf("omp_get_level() = %d\n", omp_get_level());
```

```
        printf("level: %d,"
               "number of threads in the team - %d,"
               "thread: %d\n",
               level,
               omp_get_num_threads(),
               omp_get_thread_num());

        double a[M*K] = {1};
        double b[K*M] = {1};
        double c[M*N] = {1};

        mkl_set_num_threads_local(4);
        cblas_dgemm(CblasColMajor, CblasNoTrans, CblasNoTrans,
                    M, N, K, 1.0, a, M, b, K, 1.0, c, M);
        printf("c[0] = %lf\n", c[0]);
    }
}

int main()
{
    omp_set_dynamic(0);
    omp_set_nested(1);
    omp_set_num_threads(16);
    printf("total threads: %d\n",omp_get_max_threads());
    #pragma omp parallel num_threads(2)
    {
        //omp region - 2 thread
        report_num_threads(omp_get_thread_num());
    }
    return 0;
}

int main()
{
    omp_set_dynamic(0);
    omp_set_nested(1);
    omp_set_num_threads(16);
    printf("total threads: %d\n",omp_get_max_threads());
    #pragma omp parallel num_threads(2)
    {
        //omp region - 2 thread
        report_num_threads(omp_get_thread_num());
    }
    return 0;
}
```

编译器选项为：

```
MKLROOT=/path/to/mkl/root
gcc main.c  -Wl,--start-group
${MKLROOT}/lib/intel64/libmkl_intel_ilp64.a
${MKLROOT}/lib/intel64/libmkl_gnu_thread.a
${MKLROOT}/lib/intel64/libmkl_core.a -Wl,--end-group -lgomp -
lpthread -lm -ldl -DMKL_ILP64  -m64  -I"${MKLROOT}/include"
```

输出为：

```
total threads: 16
omp_get_level() = 0
level: 0, number of threads in the team - 1, thread: 0
c[0] = 2.000000
```

可以看到没有用到多线程。

经分析，需要在编译选项中加上`-fopenmp`，可以按意图执行多线程。

```
MKLROOT=/path/to/mkl/root
gcc main.c  -Wl,--start-group
${MKLROOT}/lib/intel64/libmkl_intel_ilp64.a
${MKLROOT}/lib/intel64/libmkl_gnu_thread.a
${MKLROOT}/lib/intel64/libmkl_core.a -Wl,--end-group -lgomp -
fopenmp -lpthread -lm -ldl -DMKL_ILP64  -m64  -
I"${MKLROOT}/include"
```

输出为：

```
total threads: 16
omp_get_level() = 2
level: 0, number of threads in the team - 2, thread: 1
omp_get_level() = 2
level: 0, number of threads in the team - 2, thread: 0
omp_get_level() = 2
level: 1, number of threads in the team - 2, thread: 1
omp_get_level() = 2
level: 1, number of threads in the team - 2, thread: 0
c[0] = 2.000000
c[0] = 2.000000
c[0] = 2.000000
c[0] = 2.000000
```

补充说明：如果需要用到nested omp并行，可用`mkl_set_num_threads_local` API来实现，参考示例。