# oneAPI 技术创新奖 - 使用说明

## 大赛专属CPU及GPU协同计算

我们使用了`SYCL`编程，同时使用CPU以及GPU，进行协同计算：赛题的核心计算部分时矩阵乘运算，我们将矩阵分块，分配给两个`GPU`及CPU，进行协同计算

- 编译结果截屏：

```
PAC20226889@:~/wmk/final_round/thundersvm$ ./build_gpu.sh
##/opt/intel/oneapi/mkl/2022.1.0##
-- The C compiler identification is GNU 9.4.0
-- The CXX compiler identification is Clang 14.0.0
-- Check for working C compiler: /usr/bin/cc
-- Check for working C compiler: /usr/bin/cc -- works
-- Detecting C compiler ABI info
-- Detecting C compiler ABI info - done
-- Detecting C compile features
-- Detecting C compile features - done
-- Check for working CXX compiler: /opt/intel/oneapi/compiler/2022.1.0/linux/bin/dpcpp
-- Check for working CXX compiler: /opt/intel/oneapi/compiler/2022.1.0/linux/bin/dpcpp -- works
-- Detecting CXX compiler ABI info
-- Detecting CXX compiler ABI info - done
-- Detecting CXX compile features
-- Detecting CXX compile features - done
```

```
[ 96%] Linking CXX executable ../../bin/thundersvm-predict
[100%] Linking CXX executable ../../bin/thundersvm-train
[100%] Built target thundersvm-train
[100%] Built target thundersvm-predict
PAC20226889@:~/wmk/final_round/thundersvm$
```

- 部分`SYCL`代码展示

```
A0 = sycl::malloc_device<kernel_type>((m / 2) * k, q0);
A1 = sycl::malloc_device<kernel_type>(((m + 1) / 2) * k, q1);

B0 = sycl::malloc_device<kernel_type>(n * k, q0);
B1 = sycl::malloc_device<kernel_type>(n * k, q1);

C0 = sycl::malloc_device<kernel_type>(n * (m / 2), q0);
C1 = sycl::malloc_device<kernel_type>(n * ((m + 1) / 2), q1);

auto e0 = q0.memcpy(A0, A_host, sizeof(kernel_type) * (m / 2) * k);
auto e1 = q1.memcpy(A1, A_host + (m / 2) * k, sizeof(kernel_type) * ((m + 1) / 2) * k);
```

- `GPU`使用

```
std::vector<sycl::queue> qs;
auto platforms = sycl::platform::get_platforms();
for (auto & p : platforms)
{
    if(p.get_info<sycl::info::platform::name>().find("Level-Zero") != std::string::npos)
        continue;
    auto devices = p.get_devices();
    for(auto &d : devices)
    {
        if(d.is_gpu())
            qs.push_back(sycl::queue(d));
    }
}
```

# oneAPI 技术组件运用

## 直接编程

我们使用了`oneAPI`提供的`dpcpp`编译器`Intel® oneAPI DPC++/C++ Compiler (LLVM)`

### 编译器脚本

```
rm -rf build

CC=dpcpp

mkdir build && cd build && cmake -DCMAKE_CXX_COMPILER=$CC -DOpenMP_CXX_FLAGS="-qopenmp" -DOpenMP_CXX_LIB_NAMES="libiomp5" -DOpenMP_libiomp5_LIBRARY=/opt/intel/oneapi/compiler/latest/linux/compiler/lib/intel64_lin/libiomp5.so -DUSE_GPU=OFF -DUSE_CUDA=OFF -DUSE_PAPI=OFF -DCMAKE_MODULE_PATH=. .. && make -j

~
```

### 部分编译过程

```
-- The C compiler identification is GNU 9.4.0
-- The CXX compiler identification is Clang 14.0.0
-- Check for working C compiler: /usr/bin/cc
-- Check for working C compiler: /usr/bin/cc -- works
-- Detecting C compiler ABI info
-- Detecting C compiler ABI info - done
-- Detecting C compile features
-- Detecting C compile features - done
-- Check for working CXX compiler: /opt/intel/oneapi/compiler/2022.1.0/linux/bin/dpcpp
-- Check for working CXX compiler: /opt/intel/oneapi/compiler/2022.1.0/linux/bin/dpcpp -- works
-- Detecting CXX compiler ABI info
-- Detecting CXX compiler ABI info - done
-- Detecting CXX compile features
-- Detecting CXX compile features - done
-- Looking for pthread.h
-- Looking for pthread.h - found
-- Performing Test CMAKE_HAVE_LIBC_PTHREAD
-- Performing Test CMAKE_HAVE_LIBC_PTHREAD - Failed
-- Looking for pthread_create in pthreads
-- Looking for pthread_create in pthreads - not found
-- Looking for pthread_create in pthread
-- Looking for pthread_create in pthread - found
-- Found Threads: TRUE
-- Found OpenMP_C: -fopenmp (found version "4.5")
-- Found OpenMP_CXX: -qopenmp
-- Found OpenMP: TRUE (found version "4.5")
```

# 分析工具

## Intel® VTune™ Profiler

我们使用了`VTune`来进行程序热点分析，找到程序的执行瓶颈，进行有针对性的优化

### 使用命令

```
$ vtune -collect hotspots ./build/bin/thundersvm-train -s 0 -t 2 -g 1 -c 10 -o -1 ./data/demo
```

## 采集过程部分截图

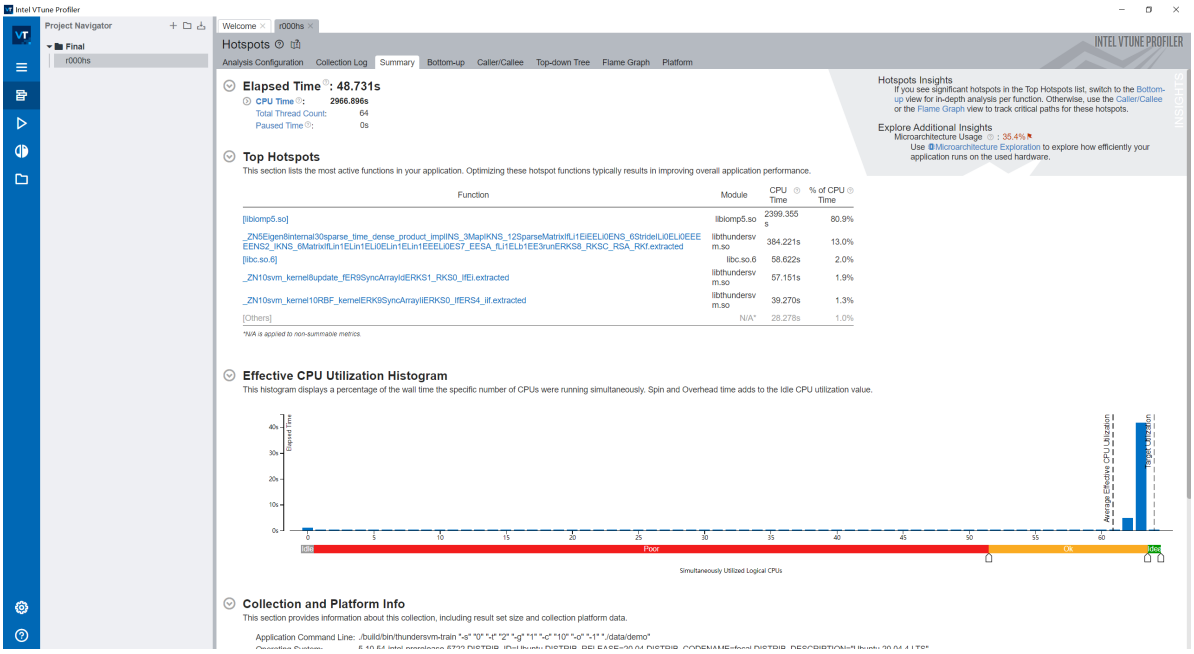
```
PAC20226889@:~/wmk/origin/thundersvm$ vtune -collect hotspots ./build/bin/thundersvm-train -s 0 -t 2 -g 1 -c 10 -o
 -1 ./data/demo
vtune: Collection started. To stop the collection, either press CTRL-C or enter from another console window: vtune
 -r /home/PAC20226889/wmk/origin/thundersvm/r000hs -command stop.
2022-11-17 09:04:06,647 INFO [default] loading dataset from file "./data/demo"
2022-11-17 09:04:07,120 INFO [default] #instances = 72309, #features = 20958
2022-11-17 09:04:07,120 INFO [default] training C-SVC
2022-11-17 09:04:07,121 INFO [default] C = 10
2022-11-17 09:04:07,121 INFO [default] gamma = 1
2022-11-17 09:04:07,142 INFO [default] #classes = 2
2022-11-17 09:04:07,240 INFO [default] working set size = 1024
2022-11-17 09:04:07,241 INFO [default] training start
2022-11-17 09:04:08,104 INFO [default] global iter = 0, total local iter = 594, diff = 2
2022-11-17 09:04:37,665 INFO [default] global iter = 100, total local iter = 42825, diff = 0.0236499
2022-11-17 09:04:54,110 INFO [default] global iter = 151, total local iter = 58981, diff = 0.000999635
2022-11-17 09:04:54,110 INFO [default] training finished
2022-11-17 09:04:54,110 INFO [default] obj = -9017.66

FOM: main loop : 46868.872710 ms,   46.868873s

2022-11-17 09:04:54,132 INFO [default] rho = -0.296003
2022-11-17 09:04:54,132 INFO [default] #sv = 19919
2022-11-17 09:04:54,359 INFO [default] #total unique sv = 19919
2022-11-17 09:04:54,365 INFO [default] training finished
vtune: Collection stopped.
vtune: Using result path `/home/PAC20226889/wmk/origin/thundersvm/r000hs'
vtune: Executing actions 19 % Resolving information for `libittnotify_collector
vtune: Warning: Cannot locate debugging information for file `/lib64/ld-linux-x86-64.so.2'.
vtune: Executing actions 19 % Resolving information for `libstdc++.so.6'
vtune: Warning: Cannot locate debugging information for file `/lib/x86_64-linux-gnu/libpthread.so.0'.
vtune: Warning: Cannot locate debugging information for file `/opt/intel/oneapi/vtune/2022.2.0/lib64/runtime/libit
tnotify_collector.so'.
vtune: Executing actions 19 % Resolving information for `libiomp5.so'
vtune: Warning: Cannot locate debugging information for file `/lib/x86_64-linux-gnu/libstdc++.so.6'.
vtune: Warning: Cannot locate debugging information for file `/lib/x86_64-linux-gnu/libc.so.6'.
vtune: Executing actions 21 % Resolving information for `libtpsstool.so'
vtune: Warning: Cannot locate debugging information for file `/opt/intel/oneapi/vtune/2022.2.0/lib64/libtpsstool.s
o'.
vtune: Executing actions 75 % Generating a report                              Elapsed Time: 48.731s
    CPU Time: 2966.896s
        Effective Time: 516.413s
        Spin Time: 2305.730s
          | A significant portion of CPU time is spent waiting. Use this metric
          | to discover which synchronizations are spinning. Consider adjusting
          | spin wait parameters, changing the lock implementation (for example,
          | by backing off then descheduling), or adjusting the synchronization
          | granularity._
```

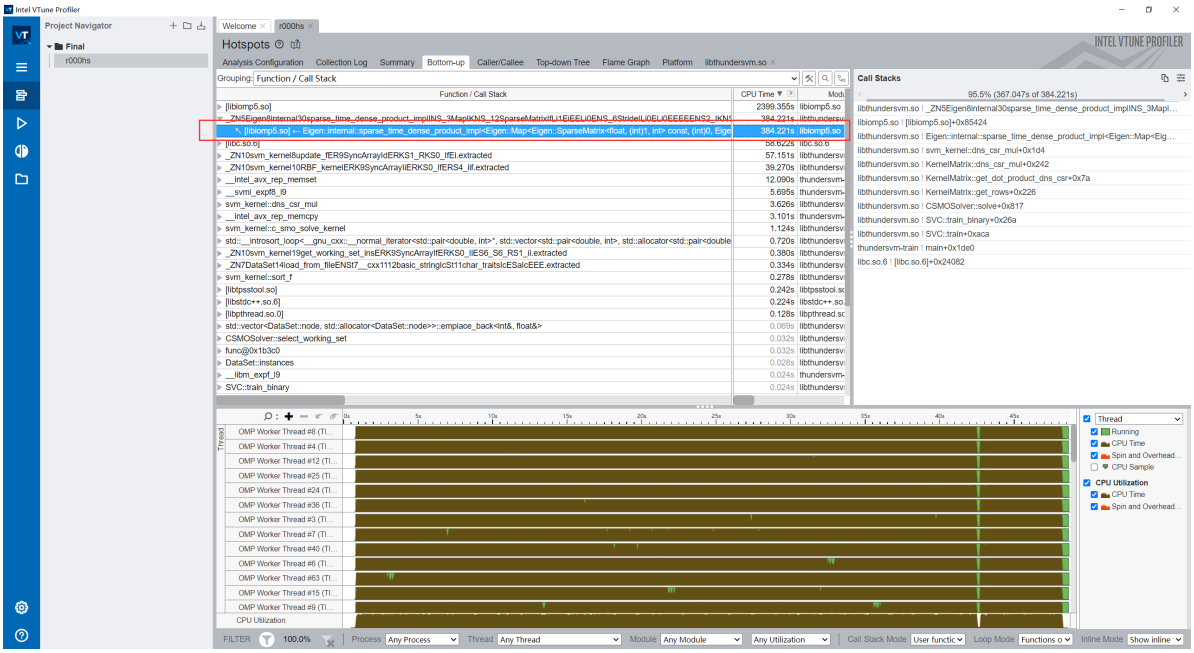
## 热点分析

通过`Summary`功能模块可以整体上看到程序的执行情况



## 程序执行瓶颈

通过`Bottom-up`功能模块可以看到当前程序执行的热点




可以看到，`Baseline`程序的最大热点在使用`Eigen`库进行矩阵乘，因此需要着重对那里进行优化

# Intel® Advisor

## CPU/Memory Roofline Insights

使用`CPU/Memory Roofline Insights`来分析硬件上的实际上限性能，并确定主要限制因素，比如是内存带宽还是计算能力

- 收集`roofline`命令

```
$ advisor --collect=roofline --project-dir=./advi  --search-dir all:r=./ --
./build/bin/thundersvm-train -s 0 -t 2 -g 1 -c 10 -o -1 ./data/demo
```

- 采集过程部分截图

```
PAC20226889@:~/wmk/origin/thundersvm$ advisor --collect=roofline --project-dir=./advi  --search-dir all:r=./ -- ./
build/bin/thundersvm-train -s 0 -t 2 -g 1 -c 10 -o -1 ./data/demo
advisor: Warning: The Roofline is a special batch mode of data collection. It runs two analyses one by one. There
are Survey Analysis and Trip Counts Analysis with FLOP respectively.
advisor: Starting command line: advisor --collect survey --project-dir ./advi --search-dir all:r=./ -- ./build/bin
/thundersvm-train -s 0 -t 2 -g 1 -c 10 -o -1 ./data/demo
Intel(R) Advisor Command Line Tool
Copyright (C) 2009-2022 Intel Corporation. All rights reserved.
advisor: Collection started. To stop the collection, either press CTRL-C or enter from another console window: adv
isor -r /home/PAC20226889/wmk/origin/thundersvm/advi/e000/hs000 -command stop.
2022-11-17 11:52:06,499 INFO [default] loading dataset from file "./data/demo"
2022-11-17 11:52:06,765 INFO [default] #instances = 72309, #features = 20958
2022-11-17 11:52:06,765 INFO [default] training C-SVC
2022-11-17 11:52:06,765 INFO [default] C = 10
2022-11-17 11:52:06,765 INFO [default] gamma = 1
2022-11-17 11:52:06,786 INFO [default] #classes = 2
2022-11-17 11:52:06,884 INFO [default] working set size = 1024
2022-11-17 11:52:06,884 INFO [default] training start
2022-11-17 11:52:07,674 INFO [default] global iter = 0, total local iter = 594, diff = 2
2022-11-17 11:52:37,939 INFO [default] global iter = 100, total local iter = 42825, diff = 0.0236499
2022-11-17 11:52:52,506 INFO [default] global iter = 151, total local iter = 58981, diff = 0.000999635
2022-11-17 11:52:52,507 INFO [default] training finished
2022-11-17 11:52:52,507 INFO [default] obj = -9017.66

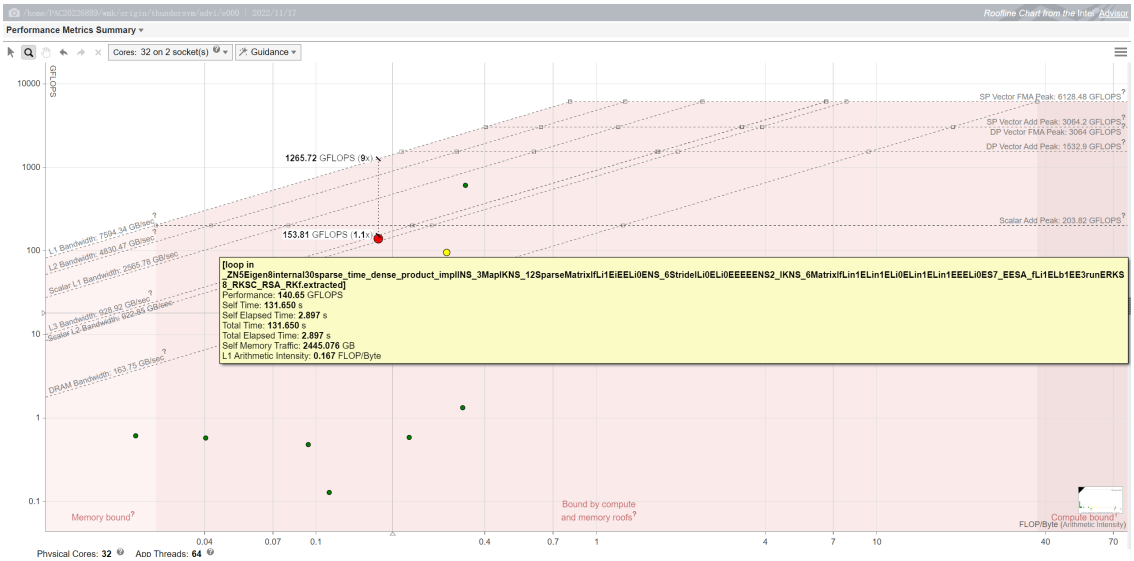
FOM: main loop : 45622.532918 ms,   45.622533s

2022-11-17 11:52:52,529 INFO [default] rho = -0.296003
2022-11-17 11:52:52,529 INFO [default] #sv = 19919
2022-11-17 11:52:52,580 INFO [default] #total unique sv = 19919
2022-11-17 11:52:52,588 INFO [default] training finished
advisor: Collection stopped.
advisor: Opening result 21 % Resolving information for `libc.so.6'
advisor: Warning: Cannot locate debugging information for file `/opt/intel/oneapi/advisor/2022.1.0/lib64/runtime/l
ibittnotify_collector.so'.
advisor: Warning: Cannot locate debugging information for file `/lib/x86_64-linux-gnu/libstdc++.so.6'.
advisor: Warning: Cannot locate debugging information for file `/lib/x86_64-linux-gnu/libc.so.6'.
advisor: Opening result 21 % Resolving information for `libtpsstool.so'
advisor: Warning: Cannot locate debugging information for file `/lib/x86_64-linux-gnu/libpthread.so.0'.
advisor: Warning: Cannot locate debugging information for file `/opt/intel/oneapi/advisor/2022.1.0/lib64/libtpssto
ol.so'.
advisor: Opening result 100 % done
advisor: Preparing frequently used data  2 % done
advisor: Preparing frequently used data 100 % done
advisor: Warning: Some target modules do not contain debug information

Program Elapsed Time: 46.56s
```

- 采集结果查看分析

```
$ advisor --report=roofline --report-output=./advi/advisor-roofline.html --
project-dir=./advi
```




  - 通过生成的`HTML`报告，我们可以看出，有一个大的红点，红点代表的操作就是我们首要考虑的优化目标，可以看到和`Vtune`分析的一致，都是因为程序中矩阵乘的效率比较低下
  - 根据红色点的位置，我们可以看出`baseline`中的矩阵乘主要受到`L3 cache`的访问速度限制，属于访存受限， 同时位于`Scalar Add Peak`之下，表名`baseline`在`CPU`上矩阵乘是标量的，应该考虑能否使用向量化

## Offload Modeling

使用`Offload Modeling`来分析`baseline`程序是否可以移植到`GPU`并从`GPU`中获益，确定最好的`offload`方法以及可能存在的性能瓶颈

- 采集命令

```
$ export SYCL_DEVICE_FILTER=opencl:cpu
$ advisor --collect=offload --config=gen11_icl --project-dir=./advi/offload
-- ./build/bin/thundersvm-train -s 0 -t 2 -g 1 -c 10 -o -1 ./data/demo
```

- 采集过程部分截图

```
PAC20226889@:~/wmk/origin/thundersvm$ advisor --collect=offload --config=gen11_icl --project-dir=./advi/offload --
 ./build/bin/thundersvm-train -s 0 -t 2 -g 1 -c 10 -o -1 ./data/demo
advisor: The 'offload' is a special batch mode for data collection. It runs several analyses one by one.
advisor: Starting command line: advisor --collect=survey --auto-finalize --static-instruction-mix --project-dir=./
advi/offload -- ./build/bin/thundersvm-train -s 0 -t 2 -g 1 -c 10 -o -1 ./data/demo
Intel(R) Advisor Command Line Tool
Copyright (C) 2009-2022 Intel Corporation. All rights reserved.
advisor: Collection started. To stop the collection, either press CTRL-C or enter from another console window: adv
isor -r /home/PAC20226889/wmk/origin/thundersvm/advi/offload/e000/hs000 -command stop.
2022-11-17 12:53:10,419 INFO [default] loading dataset from file "./data/demo"
2022-11-17 12:53:10,944 INFO [default] #instances = 72309, #features = 20958
2022-11-17 12:53:10,945 INFO [default] training C-SVC
2022-11-17 12:53:10,945 INFO [default] C = 10
2022-11-17 12:53:10,945 INFO [default] gamma = 1
2022-11-17 12:53:10,965 INFO [default] #classes = 2
2022-11-17 12:53:11,062 INFO [default] working set size = 1024
2022-11-17 12:53:11,063 INFO [default] training start
2022-11-17 12:53:12,982 INFO [default] global iter = 0, total local iter = 594, diff = 2
2022-11-17 12:53:44,054 INFO [default] global iter = 100, total local iter = 42825, diff = 0.0236499
2022-11-17 12:53:58,555 INFO [default] global iter = 151, total local iter = 58981, diff = 0.000999635
2022-11-17 12:53:58,555 INFO [default] training finished
2022-11-17 12:53:58,556 INFO [default] obj = -9017.66

FOM: main loop : 47492.599178 ms,   47.492599s

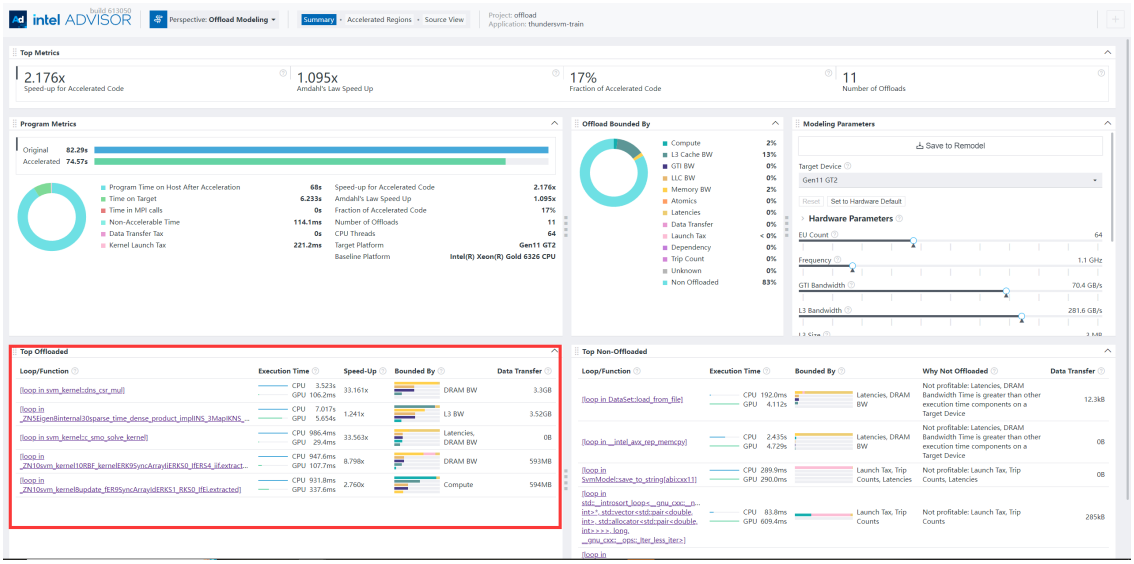
2022-11-17 12:53:58,575 INFO [default] rho = -0.296003
2022-11-17 12:53:58,575 INFO [default] #sv = 19919
2022-11-17 12:53:58,622 INFO [default] #total unique sv = 19919
2022-11-17 12:53:58,628 INFO [default] training finished
advisor: Collection stopped.
advisor: Opening result 21 % Resolving information for `libiomp5.so'
advisor: Warning: Cannot locate debugging information for file `/lib64/ld-linux-x86-64.so.2'.
advisor: Warning: Cannot locate debugging information for file `/lib/x86_64-linux-gnu/libpthread.so.0'.
advisor: Warning: Cannot locate debugging information for file `/opt/intel/oneapi/advisor/2022.1.0/lib64/runtime/l
ibittnotify_collector.so'.
advisor: Warning: Cannot locate debugging information for file `/lib/x86_64-linux-gnu/libstdc++.so.6'.
advisor: Opening result 21 % Resolving information for `libtpsstool.so'
advisor: Warning: Cannot locate debugging information for file `/lib/x86_64-linux-gnu/libc.so.6'.
advisor: Warning: Cannot locate debugging information for file `/opt/intel/oneapi/advisor/2022.1.0/lib64/libtpssto
ol.so'.
advisor: Opening result 100 % done
advisor: Preparing frequently used data  2 % done
advisor: Preparing frequently used data 100 % done
```

- 采集结果查看和分析

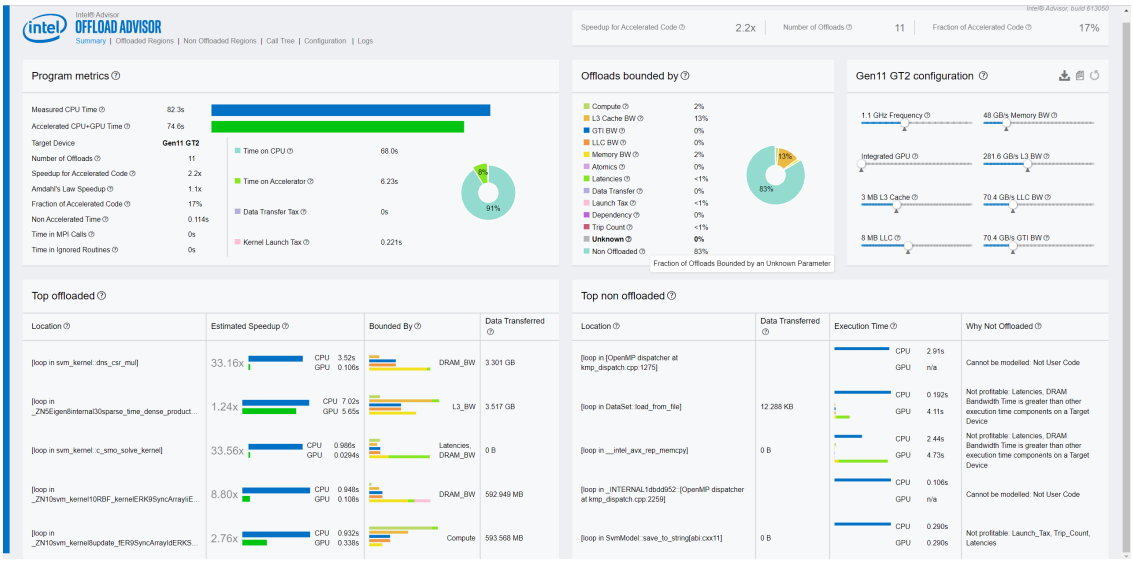
```
Measured CPU Time: 82.295s    Accelerated CPU+GPU Time: 74.568s
Speedup for Accelerated Code: 2.2x    Number of Offloads: 11    Fraction of Accelerated Code: 17%

Top Offloaded Regions
-------------------------------------------------------------------------------------------------------------------------------
-------------------------------------
 Location                                                | CPU          | GPU          | Estimated Speedup | Bound
ed By             | Data Transferred
-------------------------------------------------------------------------------------------------------------------------------
-------------------------------------
 [loop in svm_kernel::dns_csr_mul]                       |       3.523s |       0.106s |            33.16x | DRAM_
BW                |          3.301GB
 [loop in _ZN5Eigen8internal30sparse_time_dense_produ... |       7.017s |       5.654s |             1.24x | L3_BW
                   |          3.517GB
 [loop in svm_kernel::c_smo_solve_kernel]                |       0.986s |       0.029s |            33.56x | Laten
cies, DRAM_BW     |          0.000B
 [loop in _ZN10svm_kernel10RBF_kernelERK9SyncArrayIiE... |       0.948s |       0.108s |             8.80x | DRAM_
BW                |        592.949MB
 [loop in _ZN10svm_kernel8update_fER9SyncArrayIdERKS1... |       0.932s |       0.338s |             2.76x | Compu
te                |        593.568MB
-------------------------------------------------------------------------------------------------------------------------------
-------------------------------------
advisor: Opening result 25 % done
advisor: Preparing frequently used data  2 % done
advisor: Finalizing results 65 % Precomputing frequently used data
advisor: Warning: Cannot find data to precompute. Skipping the precomputation step.
advisor: Finalizing results 99 % done
advisor: Finalizing results 99 % done
advisor:   0 % done
advisor:   0 % done
advisor: Issues progress 33 % done
advisor: Issues progress  0 % done
advisor: Issues progress  0 % done
advisor: The report is saved in '/home/PAC20226889/wmk/origin/thundersvm/advi/offload/e000/report/advisor-report.h
tml'.
```

使用命令行进行`Offload Modeling`可以得到相应的`HTML`报告




上面两张图展示了使用`offload modeling`分析结果

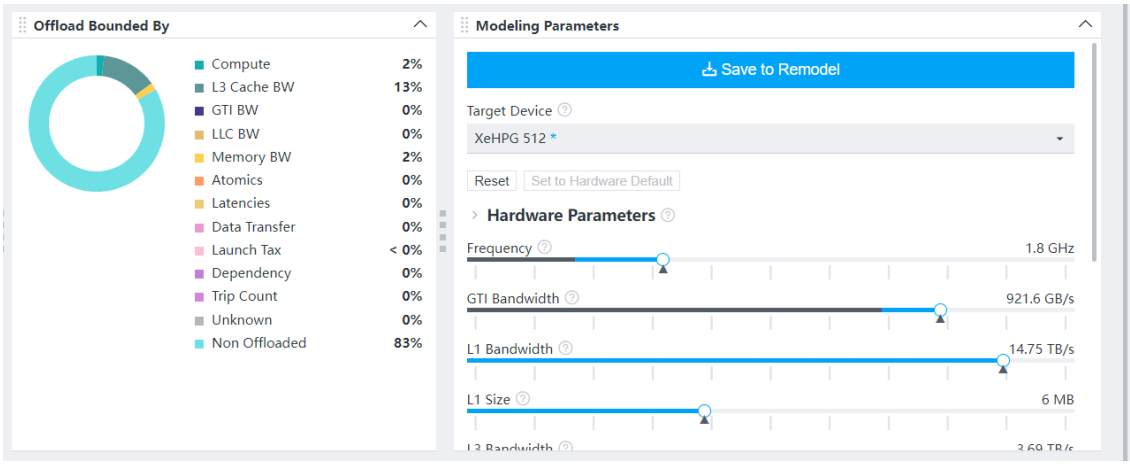


Top offloaded

| Location | Estimated Speedup | Bounded By | Data Transferred |
|---|---|---|---|
| [loop in svm_kernel::dns_csr_mul] | 33.16x CPU 3.52s GPU 0.106s | DRAM_BW | 3.301 GB |
| [loop in _ZN5Eigen8internal30sparse_time_dense_product... | 1.24x CPU 7.02s GPU 5.65s | L3_BW | 3.517 GB |
| [loop in svm_kernel::c_smo_solve_kernel] | 33.56x CPU 0.986s GPU 0.0294s | Latencies, DRAM_BW | 0 B |
| [loop in _ZN10svm_kernel10RBF_kernelERK9SyncArrayliE... | 8.80x CPU 0.948s GPU 0.108s | DRAM_BW | 592.949 MB |
| [loop in _ZN10svm_kernel8update_fER9SyncArrayIdERKS... | 2.76x CPU 0.932s GPU 0.338s | Compute | 593.568 MB |

- 上图是`Top Offloaded`展示了`offload GPU`能够获得更好的性能的部分

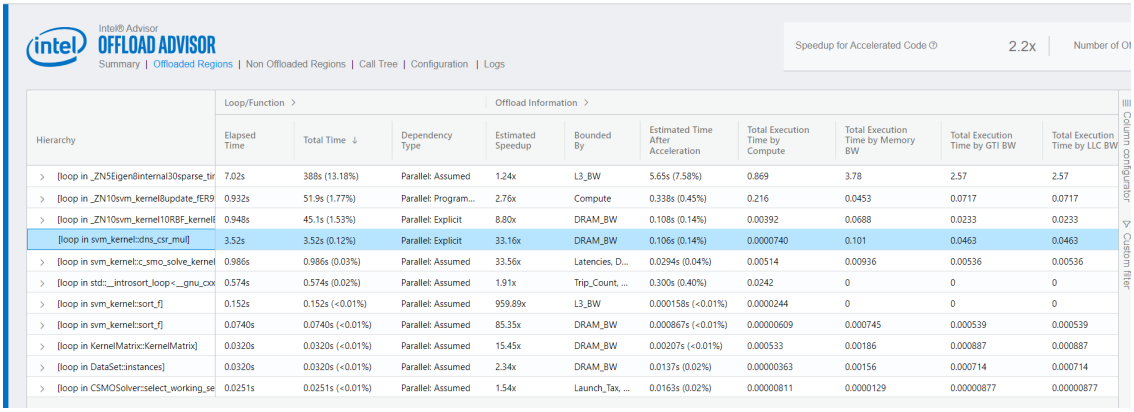
- 可以看到`Offload Modeling`给出了`baseline`中各个函数在移植到`GPU`上时预估的加速比，之前分析到的热点矩阵乘可以达到30多倍，说明移植到`GPU`上可以获得不错的优化

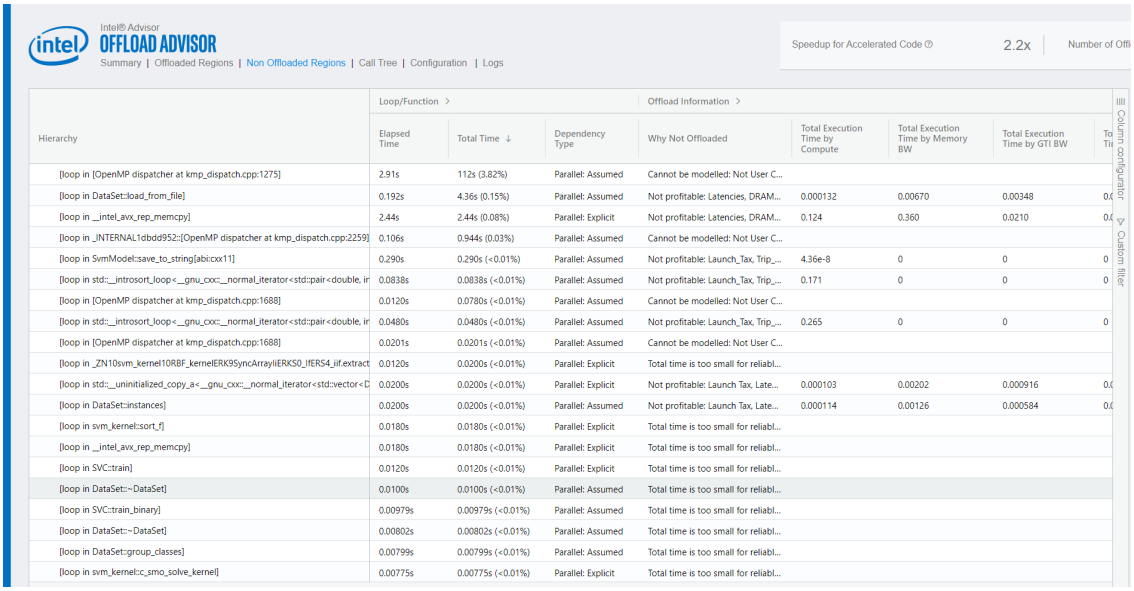



- 可以看到移植到`GPU`上时影响性能的主要因素：

  有很大一部分原因是部分代码无法`offload`，还有就是受到缓存的限制

- 下面两张图展示了能够`offload`以及不能`offload`的区域






## GPU Roofline Insights

使用`GPU Roofline Insights`来分析`GPU Offload`之后的代码的执行效率，找到限制`kernel`的主要瓶颈，获得一些优化建议，并将内核性能与GPU的限制以Roofline图表的形式可视化

- 采集指令

```
$  advisor --collect=roofline --profile-gpu --project-dir=./advi  --search-dir all:r=./ -- ./build/bin/thundersvm-train -s 0 -t 2 -g 1 -c 10 -o -1 ./data/demo
```

- 采集过程部分截图

```
advisor: Opening result 100 % done
advisor: Preparing frequently used data  9 % done
advisor: Preparing frequently used data 100 % done
advisor: Warning: Some target modules do not contain debug information
advisor: Warning: Too small execution time on GPU
advisor: Warning: No GPU Hardware Metrics Collected

Program Elapsed Time: 206.21s

CPU Time: 203.67s
Time in 15 Vectorized Loops: 126.17s
GFLOPS: 0.75
GINTOPS: 0.25

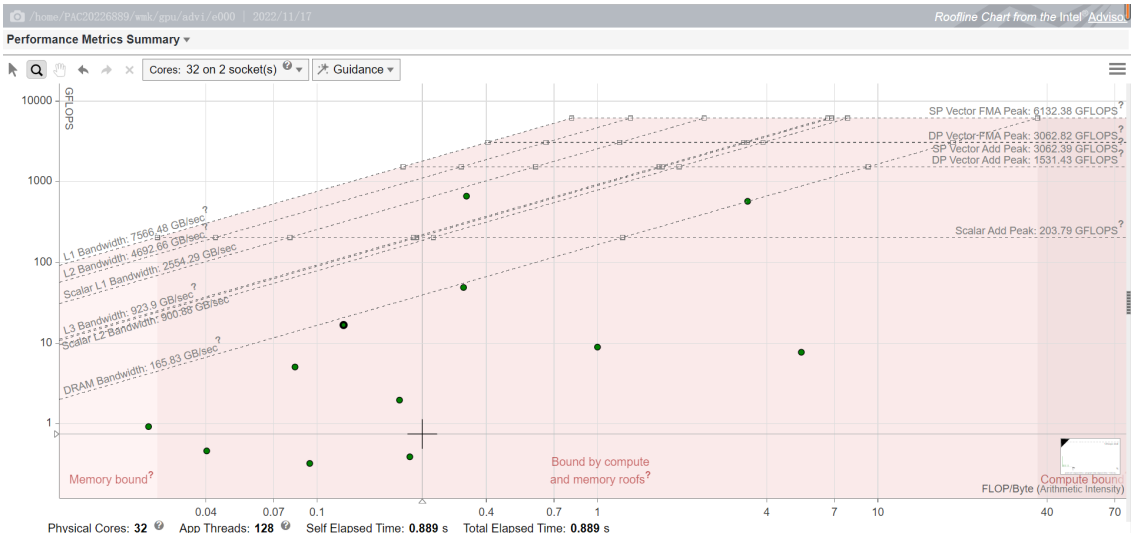
GPU Time: 2.54s
Data Transfer Time: 1.35s
EU Array Active / Stalled / Idle: 0% / 0% / 0%

Top GPU Hotspots:
           Kernel              GFLOPS    GINTOPS   Calls
-----------------------------------------------------------------
oneapi::mkl::sparse::gpu::s...   239.800   109.808   1,240
oneapi::mkl::sparse::gpu::s...     0.000   348.481   1,240

advisor: The report is saved in '/home/PAC20226889/wmk/gpu/advi/e000/trc000/advisor-survey.txt'.
advisor: Starting command line: advisor --collect=projection --enforce-baseline-decomposition --model-baseline-gpu --gpu --no-show-report --project-dir=./advi --search-di
r=all:r=./ -- ./build/bin/thundersvm-train -s 0 -t 2 -g 1 -c 10 -o -1 ./data/demo
Intel(R) Advisor Command Line Tool
Copyright (C) 2009-2022 Intel Corporation. All rights reserved.
advisor: Opening result 25 % done
advisor: Preparing frequently used data  9 % done
advisor: Preparing frequently used data 100 % done                          PAC20226889@:~/wmk/gpu$ _
```

- 生成查看报告

```
$ advisor --report=roofline --profile-gpu --report-output=./advi/advisor-roofline.html --project-dir=./advi
```




- 上图中的点表示`kernel`点的颜色和大小表示各个`kernel`之前相对的执行时间，一般来说，绿色的较小的点优化的空间比价小
- 可以看到上午中，所有的点基本大小相同，颜色都是绿色，也就说当前`gpu offload`各个`kernel`执行时间基本一致，没有很突出的热点
- 同时可以看出`kernel`受到访存的限制

## Intel® Distribution for GDB

在优化过程中，将`baseline`进行`GPU Offload`时，我们遇到一些问题，为了更好的定位问题出现的位置，我们使用了`oneapi-gdb`进行调试。`Intel® Distribution for GDB`能够处理每个设备上同时运行的数千个线程

- 使用案例

  使用`GPU Offload`，在一次换了矩阵的存储和计算格式后，运行时出现段错误

```
2022-11-17 20:28:38,274 FATAL [default] Native API failed. Native API returns: -5 (CL
_OUT_OF_RESOURCES) -5 (CL_OUT_OF_RESOURCES)
2022-11-17 20:28:38,274 WARNING [default] Aborting application. Reason: Fatal log at
[/home/PAC20226889/wmk/final_round/thundersvm/src/thundersvm/thundersvm-train.cpp:120
]
./run-data1.sh: line 1: 84481 Aborted                 (core dumped) ../build/bin/thun
dersvm-train -s 0 -t 2 -g 1 -c 10 -o -1 ../data/data1
```

  - 更改编译选项，添加`-g`，并重新编译
  - ```
    $ gdb-oneapi ./build/bin/thundersvm-train
    ```

```
PAC20226889@:~/wmk/final_round/thundersvm$ gdb-oneapi ./build/bin/thundersvm-train
GNU gdb (Intel(R) Distribution for GDB* 2022.2) 11.2
Copyright (C) 2022 Free Software Foundation, Inc.; (C) 2022 Intel Corp.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.
Type "show copying" and "show warranty" for details.
This GDB was configured as "x86_64-pc-linux-gnu".
Type "show configuration" for configuration details.

For information about how to find Technical Support, Product Updates,
User Forums, FAQs, tips and tricks, and other support information, please visit:
<http://www.intel.com/software/products/support/>.
For help, type "help".
Type "apropos word" to search for commands related to "word"...
Reading symbols from ./build/bin/thundersvm-train...
```
```

- ```
  $ set args  -s 0 -t 2 -g 1 -c 10 -o -1 ./data/data1
  $ r
  ```

```
2022-11-18 00:59:09,932 FATAL [default] Native API failed. Native API returns: -5 (CL_OUT_OF_RESOURCES) -5 (CL_OU
T_OF_RESOURCES)
2022-11-18 00:59:09,932 WARNING [default] Aborting application. Reason: Fatal log at [/home/PAC20226889/wmk/final
_round/thundersvm/src/thundersvm/thundersvm-train.cpp:120]

Thread 1 "thundersvm-trai" received signal SIGABRT, Aborted.
0x00007fffc7afa00b in raise () from /lib/x86_64-linux-gnu/libc.so.6
(gdb) _
```

- ```
  $ bt
  ```

```
(gdb) bt
#0  0x00007fffc7afa00b in raise () from /lib/x86_64-linux-gnu/libc.so.6
#1  0x00007fffc7ad9859 in abort () from /lib/x86_64-linux-gnu/libc.so.6
#2  0x00007ffff7f57a46 in el::base::utils::abort (status=<optimized out>, reason=...)
    at /home/PAC20226889/wmk/final_round/thundersvm/src/thundersvm/util/log.cpp:38
#3  0x00007ffff7f575fe in el::base::Writer::triggerDispatch (this=0x7fffffffc138)
    at /home/PAC20226889/wmk/final_round/thundersvm/src/thundersvm/util/log.cpp:2528
#4  0x000000000040a024 in el::base::Writer::~Writer (this=0x7fffffffc138)
    at /home/PAC20226889/wmk/final_round/thundersvm/include/thundersvm/util/log.h:3241
#5  0x0000000000409e75 in main (argc=<optimized out>, argv=<optimized out>)
    at /home/PAC20226889/wmk/final_round/thundersvm/src/thundersvm/thundersvm-train.cpp:124
```

# 基于 API 的编程

## Intel® oneAPI DPC++ Library

使用`DPC++`实现`GPU`简单高效的数据申请和拷贝

```
        //std::cout << __LINE__ << std::endl;

        Atmp[j].csr_row_ptr_host[mm] = nnzcount[j];
        //-------------------------

        Atmp[j].csr_val0 = sycl::malloc_device<kernel_type>(nnzcount[j], q0);

        //std::cout << __LINE__ << std::endl;
        Atmp[j].csr_val1 = sycl::malloc_device<kernel_type>(nnzcount[j], q1);
        //std::cout << __LINE__ << std::endl;

        Atmp[j].csr_col_ind0 = sycl::malloc_device<int>(nnzcount[j], q0);
        //std::cout << __LINE__ << std::endl;
        Atmp[j].csr_col_ind1 = sycl::malloc_device<int>(nnzcount[j], q1);
        //std::cout << __LINE__ << std::endl;

        Atmp[j].csr_row_ptr0 = sycl::malloc_device<int>(mm + 1, q0);
        //std::cout << __LINE__ << std::endl;
        Atmp[j].csr_row_ptr1 = sycl::malloc_device<int>(mm + 1, q1);
        //std::cout << __LINE__ << std::endl;
        //-------------------------

        q0.memcpy(Atmp[j].csr_val0, Atmp[j].csr_val_host, sizeof(kernel_type) * nnzcount[j]).wait();
        //std::cout << __LINE__ << std::endl;
        q1.memcpy(Atmp[j].csr_val1, Atmp[j].csr_val_host, sizeof(kernel_type) * nnzcount[j]).wait();
        //std::cout << __LINE__ << std::endl;

        q0.memcpy(Atmp[j].csr_col_ind0, Atmp[j].csr_col_ind_host, sizeof(int) * nnzcount[j]).wait();
        //std::cout << __LINE__ << std::endl;
        q1.memcpy(Atmp[j].csr_col_ind1, Atmp[j].csr_col_ind_host, sizeof(int) * nnzcount[j]).wait();
        //std::cout << __LINE__ << std::endl;

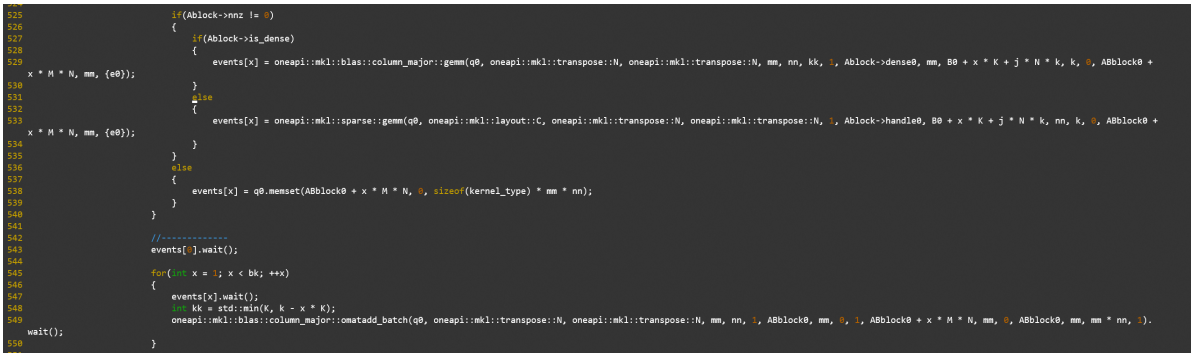
        q0.memcpy(Atmp[j].csr_row_ptr0, Atmp[j].csr_row_ptr_host, sizeof(int) * (mm + 1)).wait();
        //std::cout << __LINE__ << std::endl;
        q1.memcpy(Atmp[j].csr_row_ptr1, Atmp[j].csr_row_ptr_host, sizeof(int) * (mm + 1)).wait();
        //std::cout << __LINE__ << std::endl;

        q0.wait();
        q1.wait();


        oneapi::mkl::sparse::init_matrix_handle(&(Atmp[j].handle0));
```

## Intel® oneAPI Math Kernel Library

程序的核心计算是矩阵乘，我们使用`oneapi::mkl`中相关高性能接口



```
                    if(Ablock->nnz != 0)
                    {
                        if(Ablock->is_dense)
                        {
                            events[x] = oneapi::mkl::blas::column_major::gemm(q0, oneapi::mkl::transpose::N, oneapi::mkl::transpose::N, mm, nn, kk, 1, Ablock->dense0, mm, B0 + x * K + j * N * k, k, 0, ABblock0 +
x * M * N, mm, {e0});
                        }
                        else
                        {
                            events[x] = oneapi::mkl::sparse::gemm(q0, oneapi::mkl::layout::C, oneapi::mkl::transpose::N, oneapi::mkl::transpose::N, 1, Ablock->handle0, B0 + x * K + j * N * k, nn, k, 0, ABblock0 +
x * M * N, mm, {e0});
                        }
                    }
                    else
                    {
                        events[x] = q0.memset(ABblock0 + x * M * N, 0, sizeof(kernel_type) * mm * nn);
                    }
                }

                //------------
                events[0].wait();

                for(int x = 1; x < bk; ++x)
                {
                    events[x].wait();
                    int kk = std::min(K, k - x * K);
                    oneapi::mkl::blas::column_major::omatadd_batch(q0, oneapi::mkl::transpose::N, oneapi::mkl::transpose::N, mm, nn, 1, ABblock0, mm, 0, 1, ABblock0 + x * M * N, mm, 0, ABblock0, mm, mm * nn, 1).
wait();
                }
```

## Intel® Integrated Performance Primitives

优化过程中，我们将需要计算的矩阵进行了分块处理，每个块根据其稀疏程度采用不同的实现，最终需要将各个块的计算结果进行合并，为了提高合并的效率，采用了`Intel Integrated Performance Primitives`中的向量加法`IppStatus ippsAdd_32f_A24 (const Ipp32f* pSrc1, const Ipp32f* pSrc2, Ipp32f* pDst, Ipp32s len);`

- 代码示例

```
#include <ipp.h>
#include <ippcore.h>
#include <ippvm.h>
```

```
void MergeResult(kernel_type* dot_product,kernel_type* dot_product_sparse,int size){
//  ippsAdd_32f_A24(dot_product, dot_product_sparse, dot_product, size);
//}
  const int th_num = 16;
  int size_per_th = (size + th_num - 1) / th_num;
#pragma omp parallel for num_threads(th_num)
  for(int i = 0; i < th_num; i++)
  {
    int my_rank = omp_get_thread_num();
    int start = my_rank * size_per_th;
    int end = ((my_rank + 1) * size_per_th) > size ? size : (my_rank + 1) * size_per_th;
    int len = end > start ? end - start : 0;
    ippsAdd_32f_A24( dot_product + start, dot_product_sparse + start, dot_product + start, len );

  }
}
```

## Intel® oneAPI Threading Building Blocks

使用了`Intel® oneAPI Threading Building Blocks`的并行方法`tbb::parallel_for`

```
void RBF_kernel(const SyncArray<kernel_type> &self_dot0, const SyncArray<kernel_type> &self_dot1,
          SyncArray<kernel_type> &dot_product, int m, int n, kernel_type gamma) {
    kernel_type *dot_product_data = dot_product.host_data();
    const kernel_type *self_dot0_data = self_dot0.host_data();
    const kernel_type *self_dot1_data = self_dot1.host_data();
    tbb::parallel_for(tbb::blocked_range<size_t>(0, m), [&](tbb::blocked_range<size_t> r)
    {
        for(size_t i = r.begin(); i < r.end(); ++i)
        {
          for(int j = 0; j < n; ++j)
            dot_product_data[i * n + j] = expf(-(self_dot0_data[i] + self_dot1_data[j] - dot_product_data[i * n + j] * 2) * gamma);
        }
    });
}
```