UPPSALA
UNIVERSITET

Accelerator-Based Programming - 1TD055

ASSIGNMENT 2: PROGRAMMING IN CUDA

Jyong-Jhih Lin

September 30, 2022

0 Hardware information

snowy CPU:

```
Architecture:          x86_64
CPU op-mode(s):        32-bit, 64-bit
Byte Order:            Little Endian
CPU(s):                16
On-line CPU(s) list:   0-15
Thread(s) per core:    1
Core(s) per socket:    8
Socket(s):             2
NUMA node(s):          2
Vendor ID:             GenuineIntel
CPU family:            6
Model:                 45
Model name:            Intel(R) Xeon(R) CPU E5-2660 0 @ 2.20GHz
Stepping:              7
CPU MHz:               1200.000
CPU max MHz:           2200.0000
CPU min MHz:           1200.0000
BogoMIPS:              4388.80
Virtualization:        VT-x
L1d cache:             32K
L1i cache:             32K
L2 cache:              256K
L3 cache:              20480K
NUMA node0 CPU(s):     0-7
NUMA node1 CPU(s):     8-15
Flags:                 fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge mca cmov pat pse36 clflush dts
     acpi mmx fxsr sse sse2 ss ht tm pbe syscall nx pdpe1gb rdtscp lm constant_tsc arch_perfmon pebs bts
     rep_good nopl xtopology nonstop_tsc aperfmperf eagerfpu pni pclmulqdq dtes64 monitor ds_cpl vmx smx est
      tm2 ssse3 cx16 xtpr pdcm pcid dca sse4_1 sse4_2 x2apic popcnt tsc_deadline_timer aes xsave avx lahf_lm
      epb ssbd ibrs ibpb stibp tpr_shadow vnmi flexpriority ept vpid xsaveopt dtherm ida arat pln pts
     md_clear spec_ctrl intel_stibp flush_l1d
```

snowy memory:

```
Handle 0x1100, DMI type 17, 40 bytes
Memory Device
Array Handle: 0x1000
Error Information Handle: Not Provided
Total Width: 72 bits
Data Width: 64 bits
Size: 32 GB
Form Factor: DIMM
Set: None
Locator: PROC 1 DIMM 1
Bank Locator: Not Specified
Type: DDR3
Type Detail: Synchronous LRDIMM
Speed: 1333 MT/s
Manufacturer: HP
Serial Number: Not Specified
Asset Tag: Not Specified
Part Number: 647654-081
Rank: 4
Configured Memory Speed: 1333 MT/s
Minimum Voltage: 1.35 V
Maximum Voltage: 1.5 V
Configured Voltage: 1.35 V

#####
DDR3-1333 4-channel memory total bandwidth = 1333e6(T/s) * 64(bits) * 4(channels) / 8e9(GBytes/s)
= 42.656(GBytes/s)
```

nvidia T4, `nvidia-smi`:

```
+-----------------------------------------------------------------------------+
| NVIDIA-SMI 515.65.01    Driver Version: 515.65.01    CUDA Version: 11.7     |
|-------------------------------+----------------------+----------------------+
| GPU  Name        Persistence-M| Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp  Perf  Pwr:Usage/Cap|         Memory-Usage | GPU-Util  Compute M. |
|                               |                      |               MIG M. |
|===============================+======================+======================|
|   0  Tesla T4            On   | 00000000:08:00.0 Off |                    0 |
| N/A   30C    P8    14W /  70W |      2MiB / 15360MiB |      0%      Default |
|                               |                      |                  N/A |
+-------------------------------+----------------------+----------------------+

+-----------------------------------------------------------------------------+
| Processes:                                                                  |
|  GPU   GI   CI        PID   Type   Process name                  GPU Memory |
|        ID   ID                                                   Usage      |
|=============================================================================|
|  No running processes found                                                 |
+-----------------------------------------------------------------------------+
```

nvidia T4, *deviceQuery*:

```
/sw/EasyBuild/snowy/software/CUDA/10.1.243-iccifort-2019.5.281/extras/demo_suite/deviceQuery Starting...

 CUDA Device Query (Runtime API) version (CUDART static linking)

Detected 1 CUDA Capable device(s)

Device 0: "Tesla T4"
  CUDA Driver Version / Runtime Version          11.7 / 10.1
  CUDA Capability Major/Minor version number:    7.5
  Total amount of global memory:                 14972 MBytes (15699148800 bytes)
  (40) Multiprocessors, ( 64) CUDA Cores/MP:     2560 CUDA Cores
  GPU Max Clock rate:                            1590 MHz (1.59 GHz)
  Memory Clock rate:                             5001 Mhz
  Memory Bus Width:                              256-bit
  L2 Cache Size:                                 4194304 bytes
  Maximum Texture Dimension Size (x,y,z)         1D=(131072), 2D=(131072, 65536), 3D=(16384, 16384, 16384)
  Maximum Layered 1D Texture Size, (num) layers  1D=(32768), 2048 layers
  Maximum Layered 2D Texture Size, (num) layers  2D=(32768, 32768), 2048 layers
  Total amount of constant memory:               65536 bytes
  Total amount of shared memory per block:       49152 bytes
  Total number of registers available per block: 65536
  Warp size:                                     32
  Maximum number of threads per multiprocessor:  1024
  Maximum number of threads per block:           1024
  Max dimension size of a thread block (x,y,z): (1024, 1024, 64)
  Max dimension size of a grid size    (x,y,z): (2147483647, 65535, 65535)
  Maximum memory pitch:                          2147483647 bytes
  Texture alignment:                             512 bytes
  Concurrent copy and kernel execution:          Yes with 3 copy engine(s)
  Run time limit on kernels:                     No
  Integrated GPU sharing Host Memory:            No
  Support host page-locked memory mapping:       Yes
  Alignment requirement for Surfaces:            Yes
  Device has ECC support:                        Enabled
  Device supports Unified Addressing (UVA):      Yes
  Device supports Compute Preemption:            Yes
  Supports Cooperative Kernel Launch:            Yes
  Supports MultiDevice Co-op Kernel Launch:      Yes
  Device PCI Domain ID / Bus ID / location ID:   0 / 8 / 0
  Compute Mode:
     < Default (multiple host threads can use ::cudaSetDevice() with device simultaneously) >

deviceQuery, CUDA Driver = CUDART, CUDA Driver Version = 11.7, CUDA Runtime Version = 10.1, NumDevs = 1,
     Device0 = Tesla T4
Result = PASS
```

DELL Precision 7760 CPU:

```
Architecture:                    x86_64
CPU op-mode(s):                  32-bit, 64-bit
Byte Order:                      Little Endian
Address sizes:                   39 bits physical, 48 bits virtual
CPU(s):                          12
On-line CPU(s) list:             0-11
Thread(s) per core:              2
Core(s) per socket:              6
Socket(s):                       1
NUMA node(s):                    1
Vendor ID:                       GenuineIntel
CPU family:                      6
Model:                           141
Model name:                      Intel(R) Xeon(R) W-11855M CPU @ 3.20GHz
Stepping:                        1
CPU MHz:                         3200.000
CPU max MHz:                     4900.0000
CPU min MHz:                     800.0000
BogoMIPS:                        6374.40
Virtualization:                  VT-x
L1d cache:                       288 KiB
L1i cache:                       192 KiB
L2 cache:                        7.5 MiB
L3 cache:                        18 MiB
NUMA node0 CPU(s):               0-11
Vulnerability Itlb multihit:     Not affected
Vulnerability L1tf:              Not affected
Vulnerability Mds:               Not affected
Vulnerability Meltdown:          Not affected
Vulnerability Mmio stale data:   Not affected
Vulnerability Spec store bypass: Mitigation; Speculative Store Bypass disabled via prctl and seccomp
Vulnerability Spectre v1:        Mitigation; usercopy/swapgs barriers and __user pointer sanitization
Vulnerability Spectre v2:        Mitigation; Enhanced IBRS, IBPB conditional, RSB filling
Vulnerability Srbds:             Not affected
Vulnerability Tsx async abort:   Not affected
Flags:                           fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge mca cmov pat pse36
     clflush dts acpi mmx fxsr sse sse2 ss ht tm pbe syscall nx pdpe1gb rdtscp lm constant_tsc art
     arch_perfmon pebs bts rep_good nopl xtopology nonstop_tsc cpuid aperfmperf tsc_known_freq pni pclmulqdq
      dtes64 monitor ds_cpl vmx smx est tm2 ssse3 sdbg fma cx16 xtpr pdcm pcid sse4_1 sse4_2 x2apic movbe
     popcnt tsc_deadline_timer aes xsave avx f16c rdrand lahf_lm abm 3dnowprefetch cpuid_fault epb cat_l2
     invpcid_single cdp_l2 ssbd ibrs ibpb stibp ibrs_enhanced tpr_shadow vnmi flexpriority ept vpid ept_ad
     fsgsbase tsc_adjust bmi1 avx2 smep bmi2 erms invpcid rdt_a avx512f avx512dq rdseed adx smap avx512ifma
     clflushopt clwb intel_pt avx512cd sha_ni avx512bw avx512vl xsaveopt xsavec xgetbv1 xsaves
     split_lock_detect dtherm ida arat pln pts hwp hwp_notify hwp_act_window hwp_epp hwp_pkg_req avx512vbmi
     umip pku ospke avx512_vbmi2 gfni vaes vpclmulqdq avx512_vnni avx512_bitalg tme avx512_vpopcntdq rdpid
     movdiri movdir64b fsrm avx512_vp2intersect md_clear flush_l1d arch_capabilities
```

DELL Precision 7760 memory:

```
Handle 0x1100, DMI type 17, 92 bytes
Memory Device
    Array Handle: 0x1000
    Error Information Handle: Not Provided
    Total Width: 72 bits
    Data Width: 64 bits
    Size: 32 GB
    Form Factor: SODIMM
    Set: None
    Locator: DIMM C
    Bank Locator: BANK 0
    Type: DDR4
    Type Detail: Synchronous
    Speed: 2933 MT/s
    Manufacturer: 01980000802C
    Serial Number: 97B0B609
    Asset Tag: 04212100
    Part Number: 9965657-029.A00G
    Rank: 2
    Configured Memory Speed: 2933 MT/s
    Minimum Voltage: Unknown
    Maximum Voltage: Unknown
    Configured Voltage: 1.2 V
    Memory Technology: DRAM
    Memory Operating Mode Capability: Volatile memory
    Firmware Version: Not Specified
    Module Manufacturer ID: Bank 2, Hex 0x98
    Module Product ID: Unknown
    Memory Subsystem Controller Manufacturer ID: Unknown
    Memory Subsystem Controller Product ID: Unknown
    Non-Volatile Size: None
    Volatile Size: 32 GB
    Cache Size: None
    Logical Size: None

#####
DDR4-2933 2-channel memory total bandwidth = 2933e6(T/s) * 64(bits) * 2(channels) / 8e9(GBytes/s)
= 46.928(GBytes/s)
```

nvidia A3000, `nvidia-smi`:

```
+-----------------------------------------------------------------------------+
| NVIDIA-SMI 510.47.03    Driver Version: 510.47.03    CUDA Version: 11.6     |
|-------------------------------+----------------------+----------------------+
| GPU  Name        Persistence-M| Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp  Perf  Pwr:Usage/Cap|         Memory-Usage | GPU-Util  Compute M. |
|                               |                      |               MIG M. |
|===============================+======================+======================|
|   0  NVIDIA RTX A300...  Off  | 00000000:01:00.0  On |                  N/A |
| N/A   58C    P0    36W /  N/A |   1606MiB /  6144MiB |    100%      Default |
|                               |                      |                  N/A |
+-------------------------------+----------------------+----------------------+

+-----------------------------------------------------------------------------+
| Processes:                                                                  |
|  GPU   GI   CI        PID   Type   Process name                  GPU Memory |
|        ID   ID                                                   Usage      |
|=============================================================================|
|    0   N/A  N/A      3964      G   /usr/lib/xorg/Xorg                109MiB |
|    0   N/A  N/A     12824      G   /usr/lib/xorg/Xorg                603MiB |
|    0   N/A  N/A     12940      G   /usr/bin/gnome-shell              271MiB |
|    0   N/A  N/A     30513      G   ...308337019390783085,131072      481MiB |
|    0   N/A  N/A    215156      G   ...R2021a/bin/glnxa64/MATLAB        3MiB |
|    0   N/A  N/A    216972      G   ...GL_KHR_blend_equation_adv        5MiB |
|    0   N/A  N/A    284196      C   ./stream_triad_cuda               113MiB |
+-----------------------------------------------------------------------------+
```

nvidia A3000, `deviceQuery`:

```
./deviceQuery Starting...
```

```

 CUDA Device Query (Runtime API) version (CUDART static linking)

Detected 1 CUDA Capable device(s)

Device 0: "NVIDIA RTX A3000 Laptop GPU"
  CUDA Driver Version / Runtime Version          11.6 / 11.6
  CUDA Capability Major/Minor version number:    8.6
  Total amount of global memory:                 5913 MBytes (6200098816 bytes)
  (32) Multiprocessors, (128) CUDA Cores/MP:     4096 CUDA Cores
  GPU Max Clock rate:                            1560 MHz (1.56 GHz)
  Memory Clock rate:                             5501 Mhz
  Memory Bus Width:                              192-bit
  L2 Cache Size:                                 3145728 bytes
  Maximum Texture Dimension Size (x,y,z)         1D=(131072), 2D=(131072, 65536), 3D=(16384, 16384, 16384)
  Maximum Layered 1D Texture Size, (num) layers  1D=(32768), 2048 layers
  Maximum Layered 2D Texture Size, (num) layers  2D=(32768, 32768), 2048 layers
  Total amount of constant memory:               65536 bytes
  Total amount of shared memory per block:       49152 bytes
  Total number of registers available per block: 65536
  Warp size:                                     32
  Maximum number of threads per multiprocessor:  1536
  Maximum number of threads per block:           1024
  Max dimension size of a thread block (x,y,z): (1024, 1024, 64)
  Max dimension size of a grid size    (x,y,z): (2147483647, 65535, 65535)
  Maximum memory pitch:                          2147483647 bytes
  Texture alignment:                             512 bytes
  Concurrent copy and kernel execution:          Yes with 2 copy engine(s)
  Run time limit on kernels:                     Yes
  Integrated GPU sharing Host Memory:            No
  Support host page-locked memory mapping:       Yes
  Alignment requirement for Surfaces:            Yes
  Device has ECC support:                        Disabled
  Device supports Unified Addressing (UVA):      Yes
  Device supports Compute Preemption:            Yes
  Supports Cooperative Kernel Launch:            Yes
  Supports MultiDevice Co-op Kernel Launch:      Yes
  Device PCI Domain ID / Bus ID / location ID:   0 / 1 / 0
  Compute Mode:
     < Default (multiple host threads can use ::cudaSetDevice() with device simultaneously) >

deviceQuery, CUDA Driver = CUDART, CUDA Driver Version = 11.6, CUDA Runtime Version = 11.6, NumDevs = 1,
    Device0 = NVIDIA RTX A3000 Laptop GPU
Result = PASS
```

1 Task 1

1.1 a

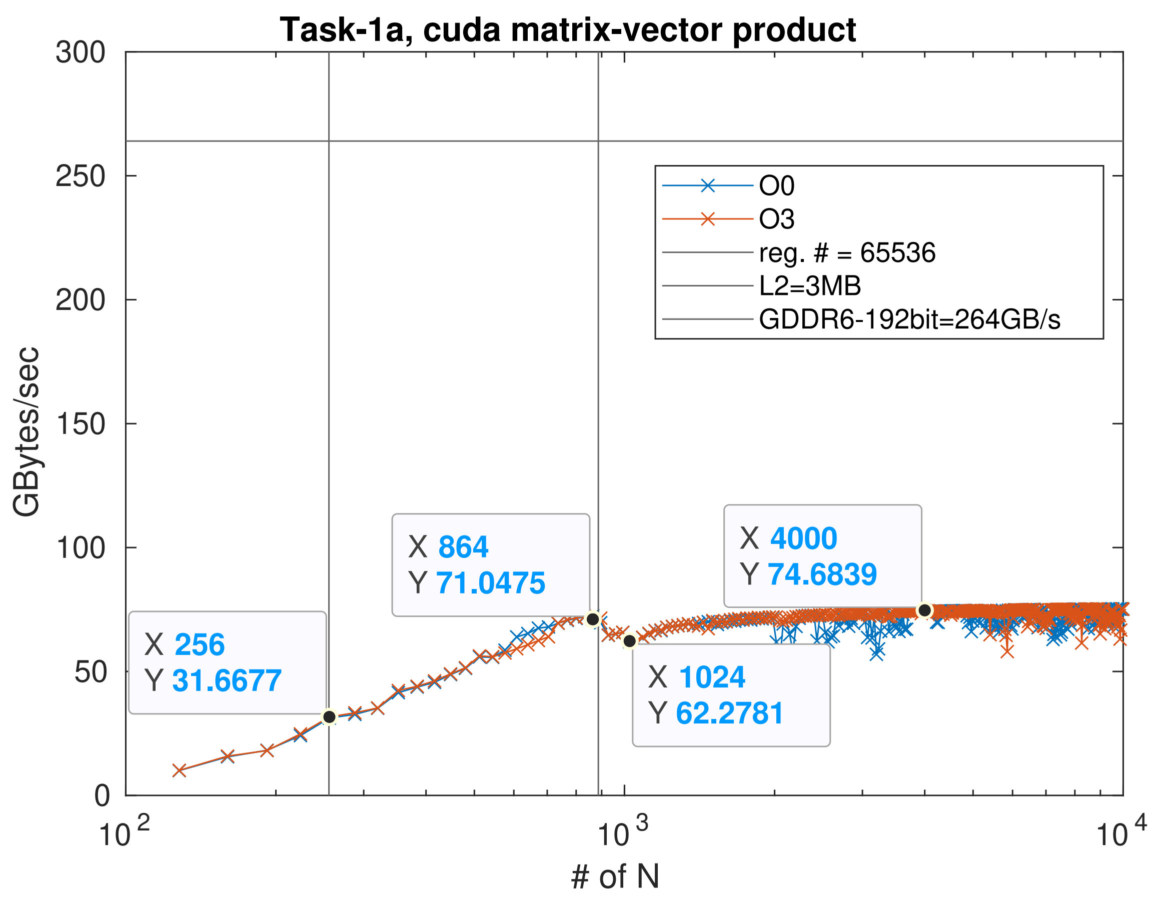

Figure 1: A3000

According to the A3000 specification, the register and the L1 cache share the same memory space which is 65536 registers. The register and L2 cache line are determined by solving "$N^2 + N = memory$".

Theoretically, a optimized matrix-vector product cuda program should be bounded by the memory bandwidth. However, in my program, the reduction algorithm is the simplest atomic-add which makes it calculation bound. The performance increases with the array length N until the L2 size.

Right after the L2 cache region, the performance has a slightly drop and soon recovers to its maximum. My hypothesis is that the drop is because of the DRAM latency and when the N is large enough, some kind of hardware prefetch functionality covers this latency therefore the performance increases back to its calculation bound.

The O0 and O3 performances are identical. It is very reasonable because a) there is basically no operations on the CPU side, b) the Ox optional is for the CPU optimization and the GPU optimization is controlled by "–gpu-architecture=compute_86 –gpu-code=sm_86" which is the same for both programs.

1.2 b

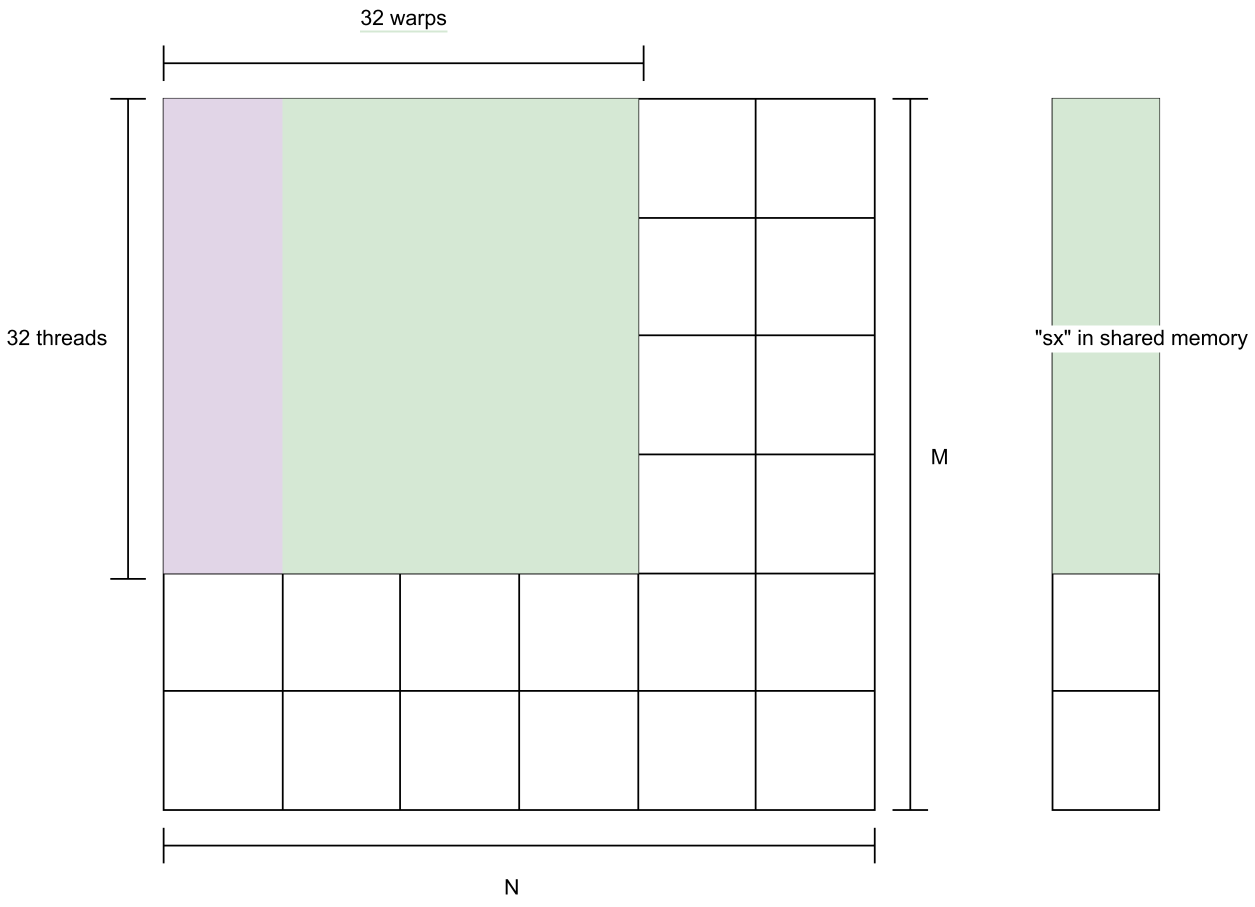

Figure 2: Parallelization algorithm

My algorithm is illustrated as the figure 2.

For A3000, there are 32 SM, 4 warp schedulers per SM, and the warp length is 32 threads. Therefore, I design a 32 by 32 thread block which gives a warp with a full 32 threads, 8 warps per warp schedule for the warp switching. It should be enough to keep a SM busy and reach its maximum SM performance. The block grid is (M/warp) by (N/warp) which could give enough blocks to occupied all 32 SM to reach its maximum total performance.

2 Task 2

2.1 a

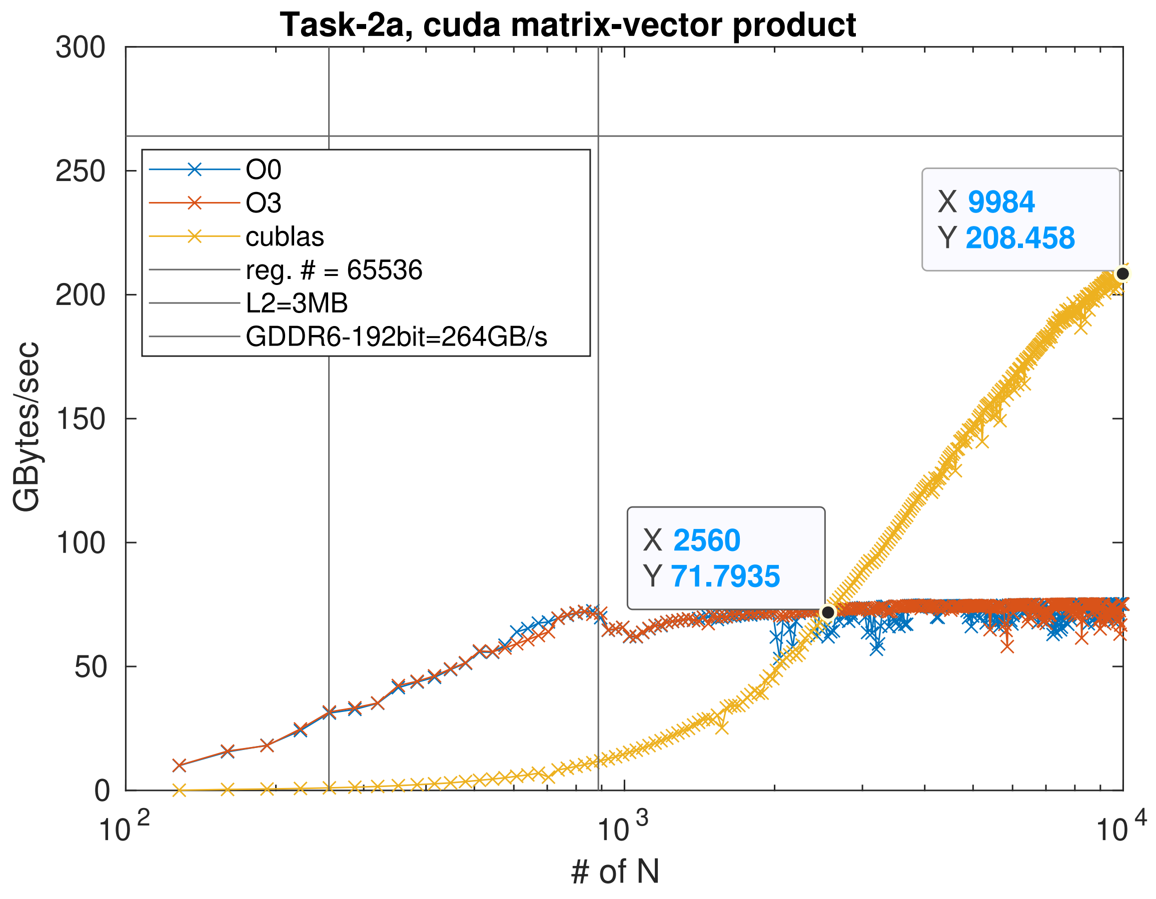

Figure 3: A3000

My program is calculation bound and limited by the reduction algorithm. For the cuBLAS library, it should suffer the memory bandwidth bound and the experiment result supports this argument. It would require an even much larger array size to test the cuBLAS library and know whether it could reach the exact GDDR6 maximum bandwidth.

My algorithm, however, has a better performance than the cuBLAS library until N=2560. It needs to know the exact implementation of the cuBLAS matrix-vector product to explain why the cuBLAS has such a low performance when the N is small.

2.2 b

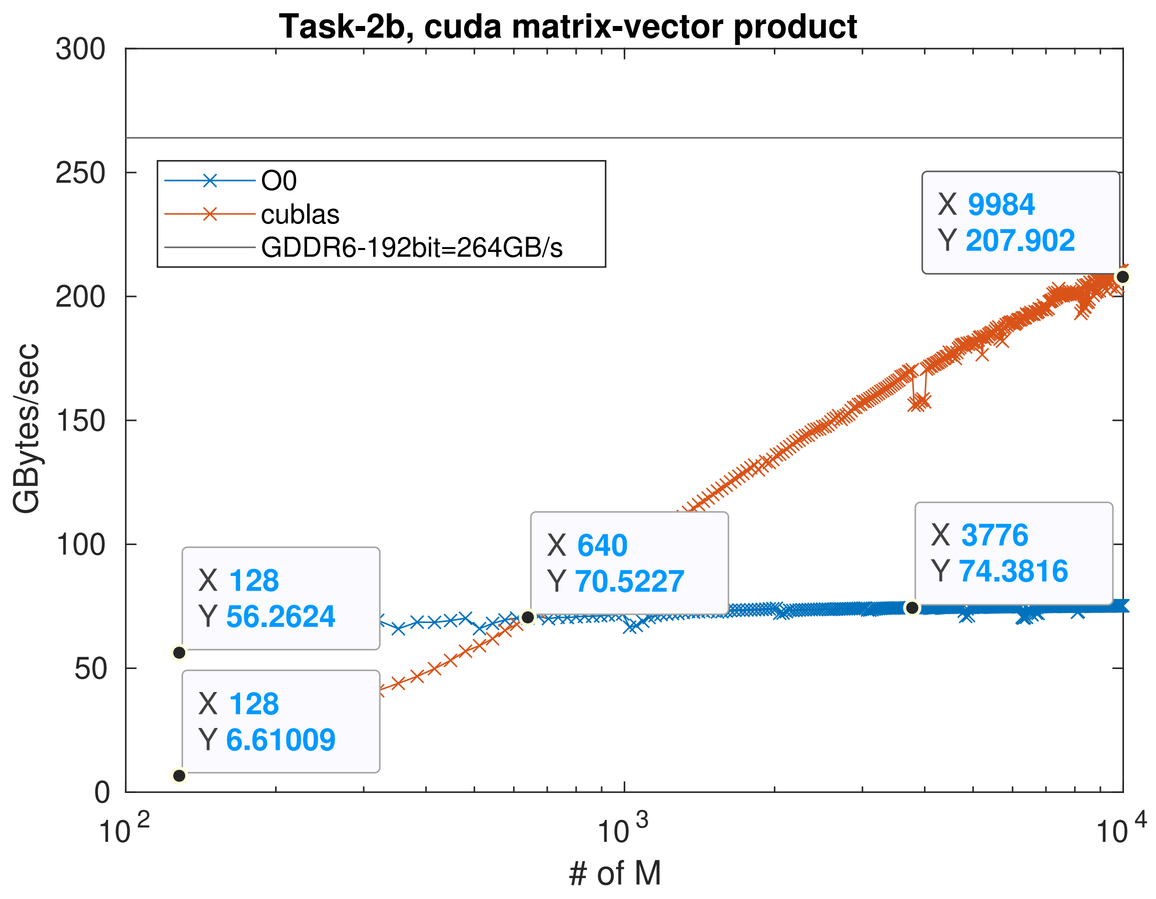

Figure 4: A3000, N=10000

2.3 c

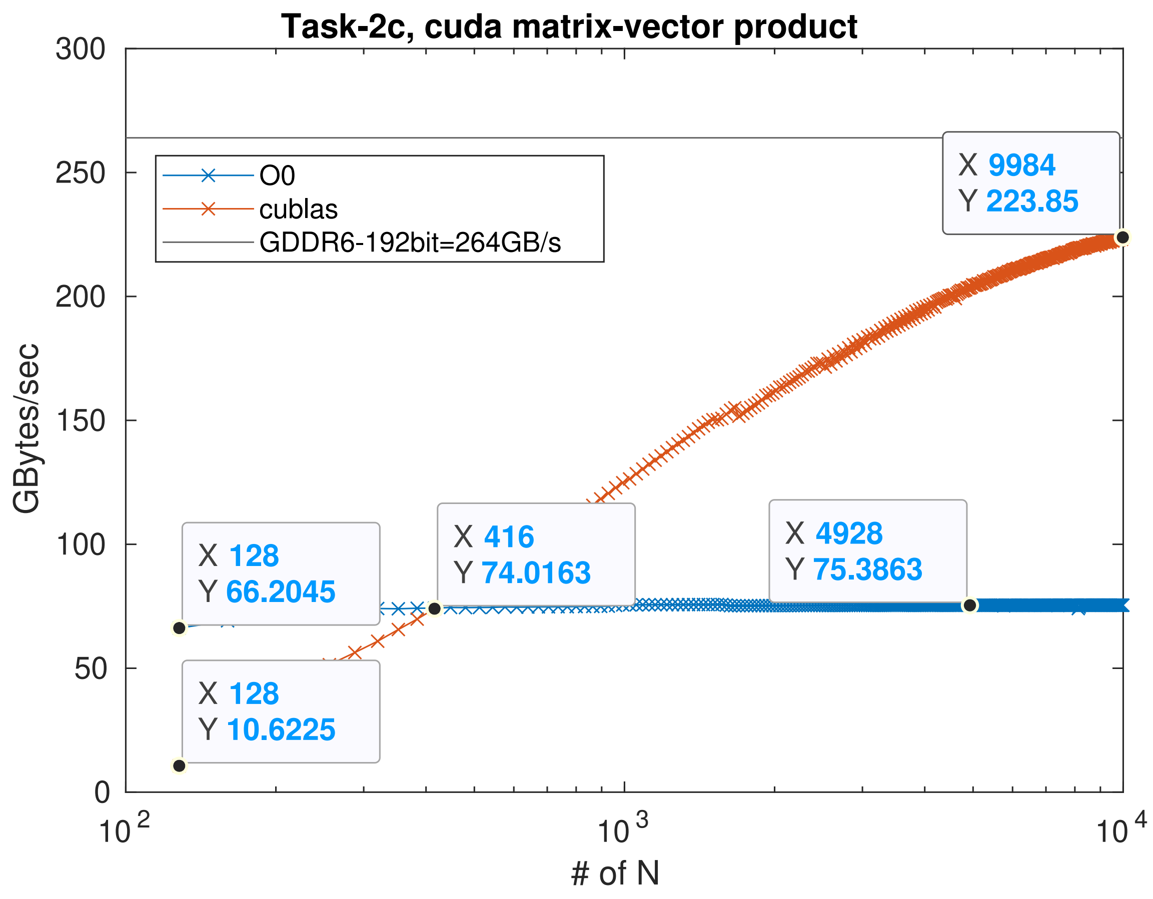

Figure 5: A3000, M=16384

In my opinion, I would say the performance behaviors in task2-b and task2-c are the same which means their parallelization algorithm do not have a preferred direction M or N. In task2-c, their performances are higher than in task2-b, and the cross-over point is earlier because the "M=16384" is larger than "N=10000". It means that the calculation amount in task2-c is larger than its corresponding point in task2-b. It agrees with the previous experiment results that my program needs enough calculation to reach it maximum performance and the cuBLAS performance is proportional to the calculation amount.

3 Task 3

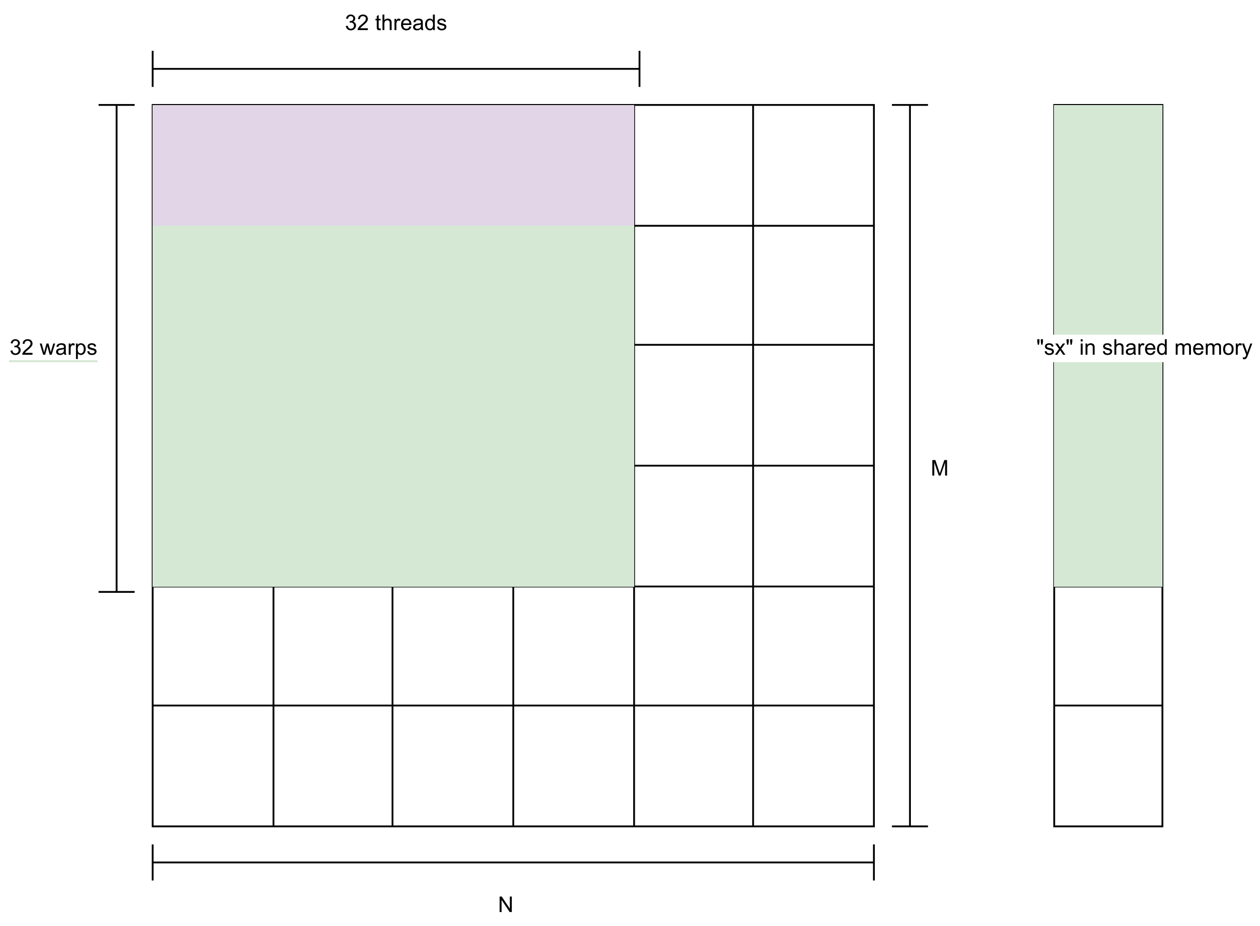

Figure 6: A3000

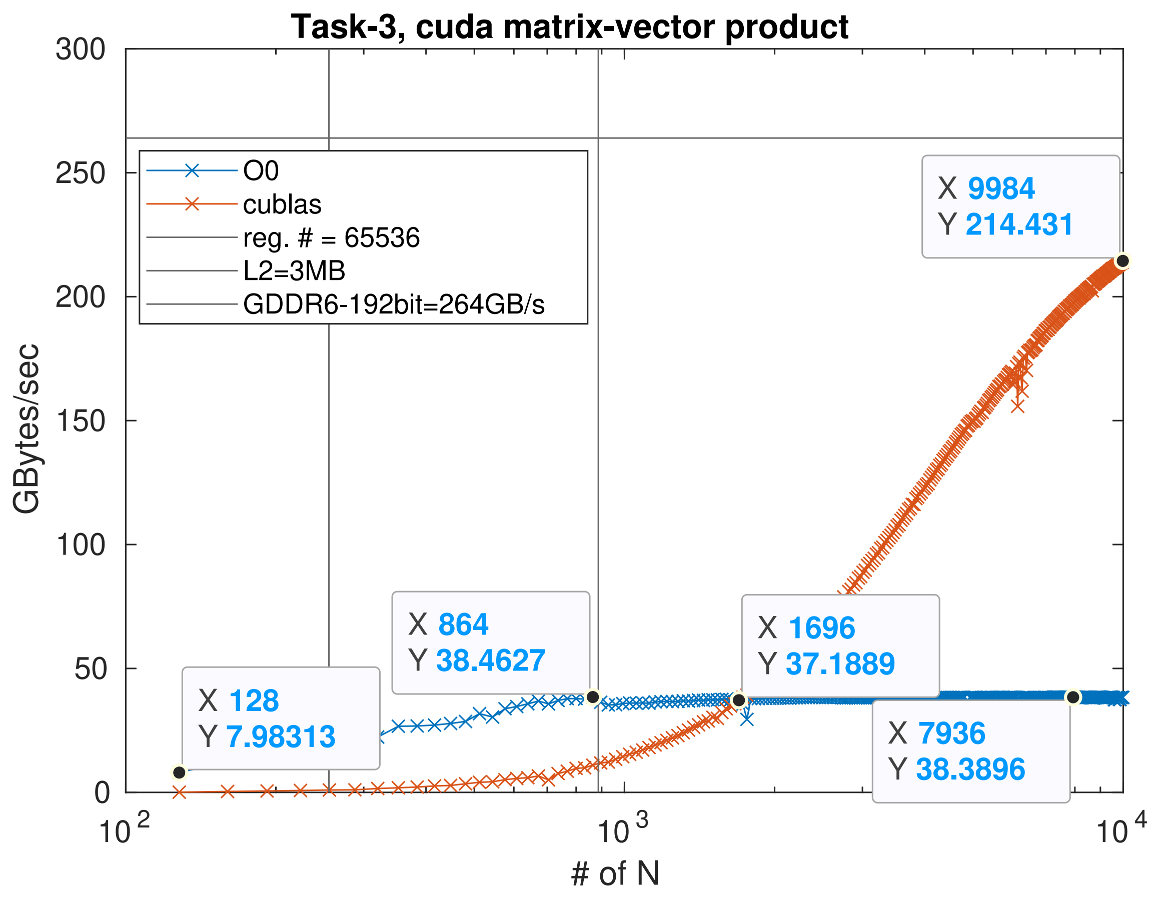

Figure 7: A3000

The cuBLAS performance and behavior are basically identical to the task2 which means that the column-major and row-major memory layout do not affect its performance. However, the performance of my program is half of the performance in task2. My algorithm is square-tile style which is not affected by the M/N ratio but the atomic-add reduction is affected by the memory layout. In my algorithm, there would be more atomic-add operations in the row-major layout than in the column-major layout.

4 Task 4

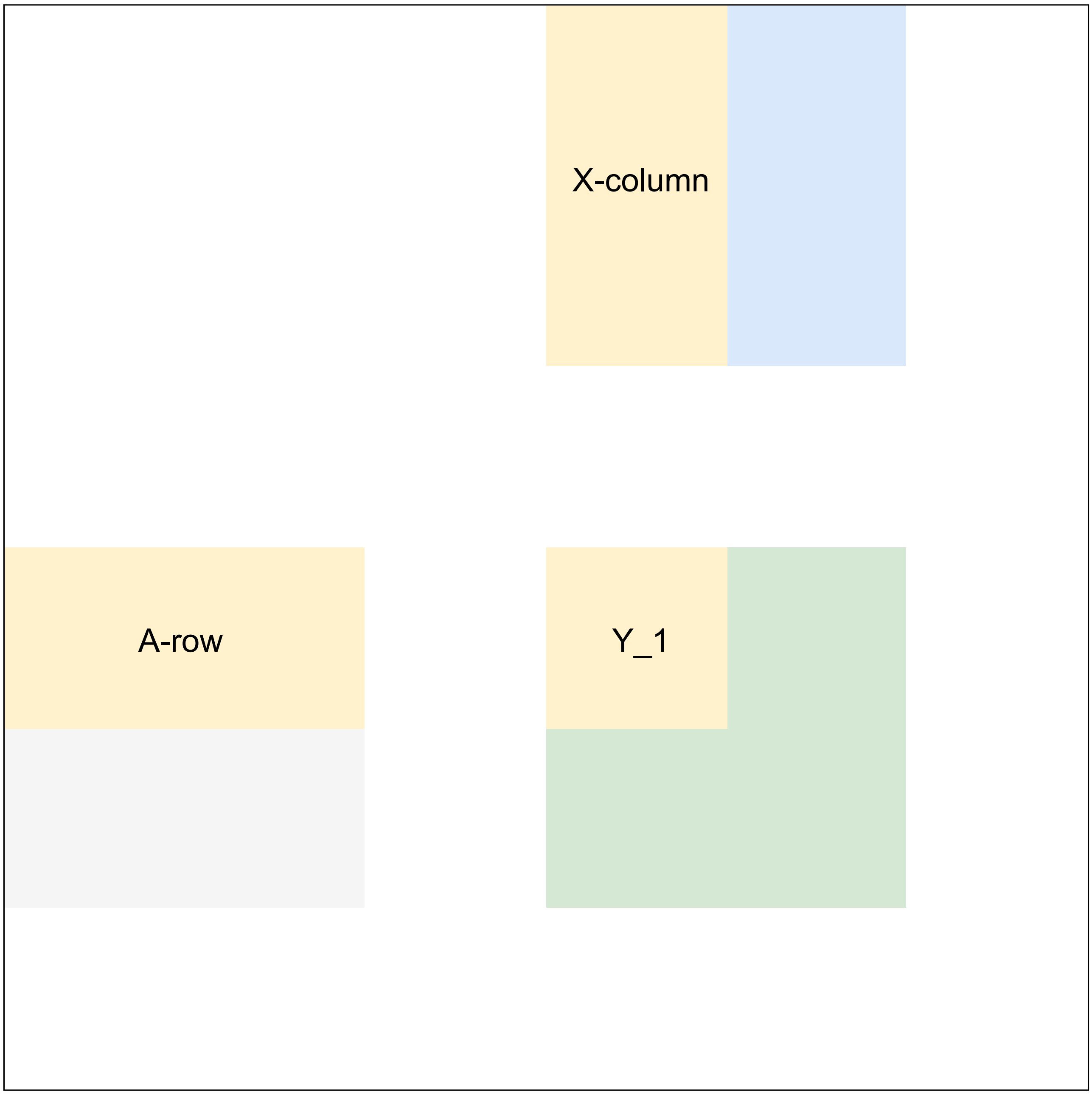

Figure 8: my algorithm

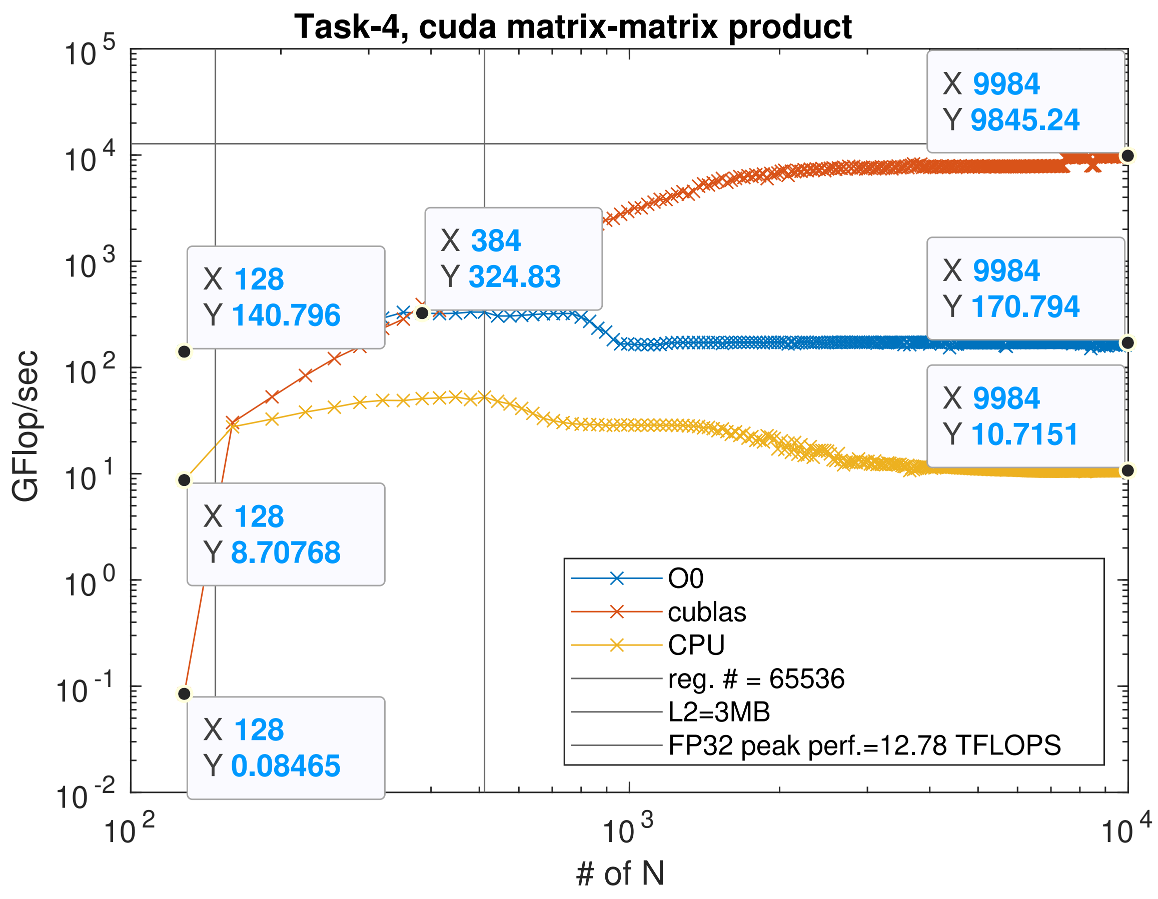

Figure 9: A3000, semilogx

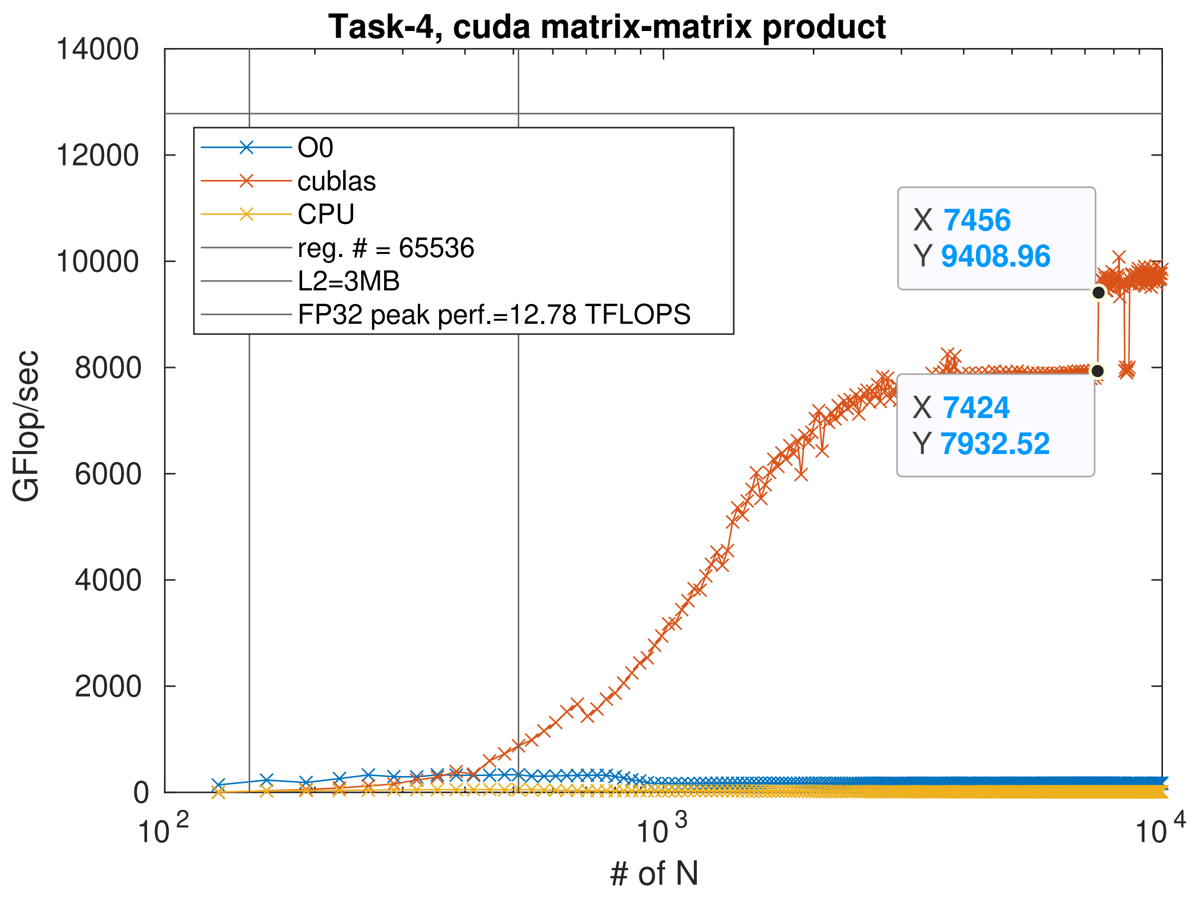

Figure 10: A3000, linear

My algorithm is illustrated as the figure 8. The CPU version is a single core and triple loop calculation. The GPU version is parallelized by putting each thread on each element of Y matrix.

For very small matrixes, which N is less than 128, the performance is "my cuda" > "my CPU" > "cuBLAS". Approximately in the L2 cache region, the performance is "my cuda" > "cuBLAS" > "my CPU". For the large matrixes, the performance is "cuBLAS" >> "my cuda" > "my CPU".

I noticed that there is a strange performance jump at N=7456 in cuBLAS, which I cannot explain. My guess is cuBLAS might change its algorithm corresponding to different matrix size?