UPPSALA
UNIVERSITET

Accelerator-Based Programming - 1TD055

ASSIGNMETN 4: KOKKOS

Jyong-Jhih Lin

October 10, 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

Listing 1: explicitly data transfer and initiation

```
    Kokkos::View<fptype*[3][3], Kokkos::memlayout, MemSpace> J( "J", n_element);
    Kokkos::View<fptype*[4][4], Kokkos::memlayout, MemSpace> A( "A", n_element);

    Kokkos::View<fptype*[3][3], Kokkos::memlayout, MemSpace>::HostMirror h_J = Kokkos::create_mirror_view( J );
    Kokkos::View<fptype*[4][4], Kokkos::memlayout, MemSpace>::HostMirror h_A = Kokkos::create_mirror_view( A );

    Kokkos::Timer timer;
    for ( int i =0; i<n_element; i++){
        h_J(i,0,0) = 3;
        h_J(i,0,1) = 1;
        h_J(i,0,2) = 1;
        h_J(i,1,0) = 1;
        h_J(i,1,1) = 3;
        h_J(i,1,2) = 1;
        h_J(i,2,0) = 1;
        h_J(i,2,1) = 1;
        h_J(i,2,2) = 3;
    }
    double t_j_init = timer.seconds();

    timer.reset();
    Kokkos::deep_copy( J, h_J );
    Kokkos::fence();
    double t_j_init_copy = timer.seconds();
    timer.reset();
    Kokkos::deep_copy( A, h_A );
    Kokkos::fence();
    double t_a_init_copy = timer.seconds();
```

Listing 2: explicitly data transfer and initiation

```
    timer.reset();
    Kokkos::deep_copy( h_J, J );
    Kokkos::fence();
    double t_j_end_copy = timer.seconds();
    timer.reset();
    Kokkos::deep_copy( h_A, A );
    Kokkos::fence();
    double t_a_end_copy = timer.seconds();
```

1.2 b

Listing 3: implementations both with float and double numbers

```
    #define fptype float
```

By changing the “fptype” definition, you can change between “float” and “double” type.

1.3 c

Too many places to be listed out. Please check the source code directly.

1.4 d

Listing 4: implementations both with float and double numbers

```
    #define memlayout LayoutLeft
```

By changing the “memlayout” definition, you can change between “LayoutLeft” and “LayoutRight”.

2 Task 2

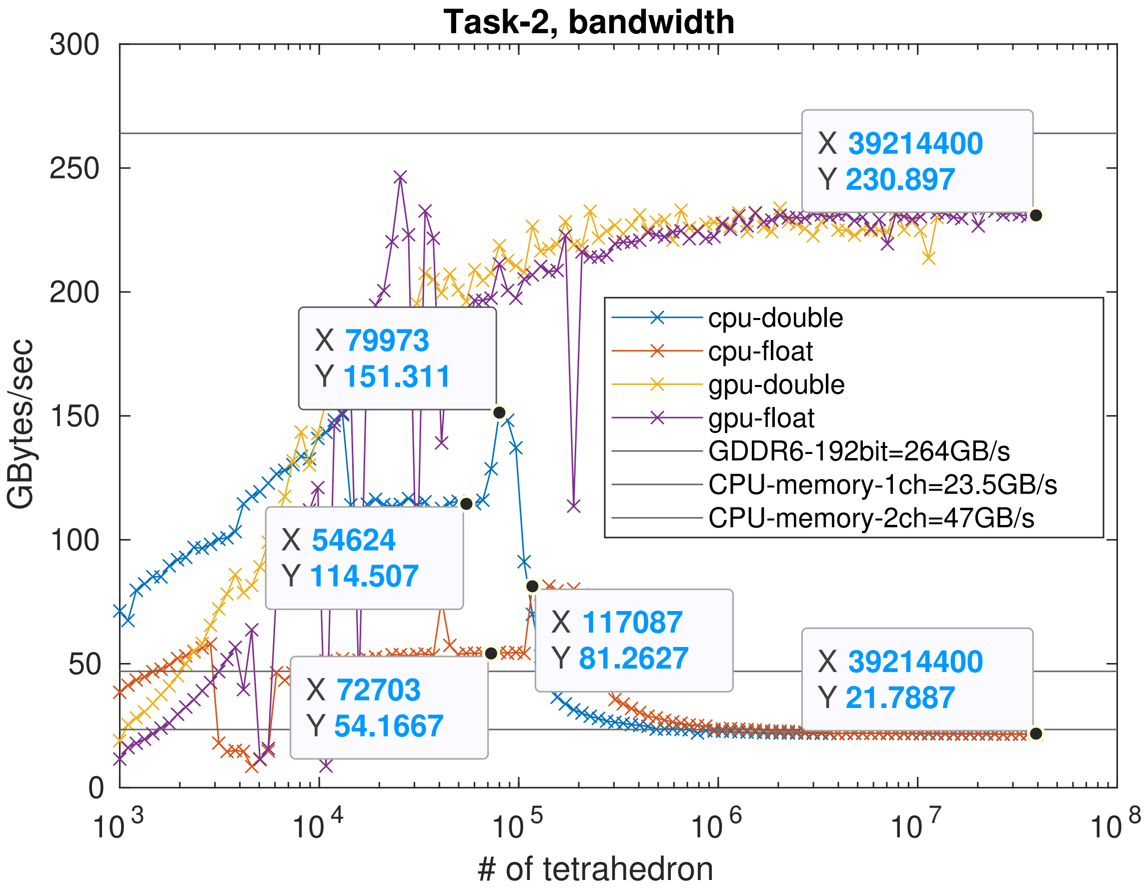

Figure 1: GPU=A3000, CPU=W-11855M

execution environment configuration:

```
OMP_NUM_THREADS=6 OMP_PROC_BIND=spread OMP_PLACES=cores ./ABP_lab4_t1.host
```

In the DRAM region, the CPU performance achieves the DDR4-2933 1-channel bandwidth as expected and is consistent with the previous lab experiments. The GPU performance is also close to the GDDR6 peak bandwidth. The CPU and GPU are both in memory bandwidth bound when the matrix size is in the DRAM region. Because they are in the memory bandwidth bound, the double and float type version have the same performance bandwidth.

The CPU version achieves its maximum performance at N around 10^5 which is in the cache region. When the matrix size is in the cache region, the cache memory bandwidth is no longer an issue and it becomes a compute bound. Therefore, the double type version has twice bandwidth performance as the float type one.

For the GPU version, because the cache size per thread is relatively much smaller than the CPU, the cache bandwidth only matters when the N is very small. Therefore, the GPU-double and GPU-float version basically have the same performance after N=10^4.

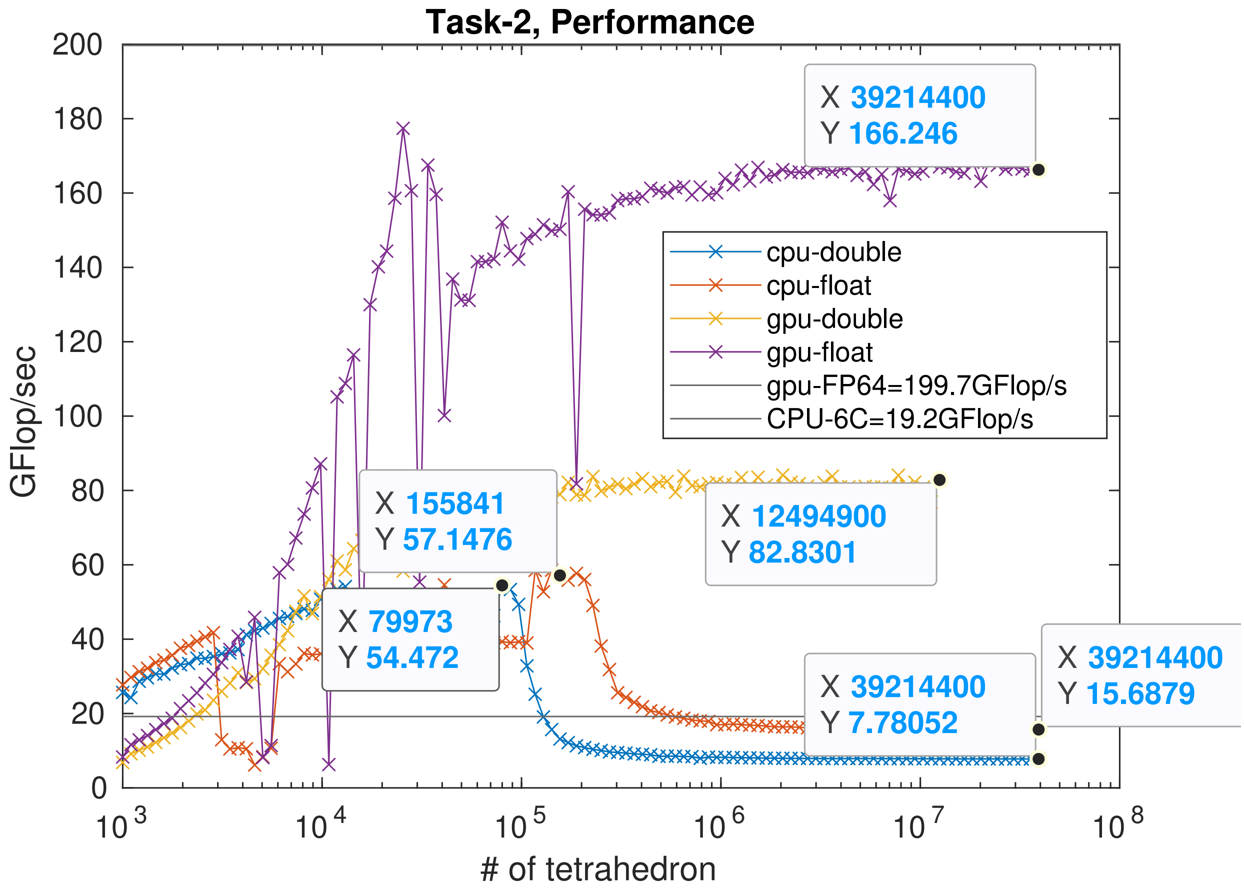

Figure 2: GPU=A3000, CPU=W-11855M

In the DRAM region, the CPU-double version has half of the compute performance of the CPU-float version because of the memory bandwidth bound. In the cache region, the CPU-double and CPU-float version have the same compute performance because the cache memory bandwidth is no longer an issue.

For the GPU version, the GPU-float version has twice compute performance as the GPU-double version because of the memory bandwidth bound, except when the N is very small and in the cache region.

3 Task 3

According to the hardware specification, `https://www.techpowerup.com/gpu-specs/rtx-a3000-mobile.c3806` , the peak compute performance and the peak memory bandwidth for NVIDIA RTX A3000 GPU are 12.78 TFLOPS for FP32, 199.7 GFLOPS for FP64, and 264.0 GB/s.

The CPU performance could be estimated by the following formula:

$$\frac{P}{GFLOP/S} = \frac{clock\ freq.}{GHz} * (\#\ of\ cores) * (SIMD\ width) * \frac{instructions}{cycle} * \frac{FLOPs}{instruction}$$

For the Intel Xeon W-11855M Processor, the clock frequency is dynamic and I will use its base frequency 3.2GHz as an estimation. It has 6 physical cores. In this program, I assume that it does not use any SIMD instructions, even with the optimization O3. Usually, the instruction per cycle is 2. I assume that the compiler does not use any FMA instructions, even with the optimization O3. Therefore, the FLOPs per instruction is 1. The final estimation of the CPU performance for the Kokkos program is 38.4 GFlop/s

For the maximum peak compute performance, the SIMD width should be 512/32 and the FLOPs per instruction should be 2 for FMA. The estimation would be 1228.8 GFlop/s.

If the compute algorithm does not change, the limiting resource for the CPU and GPU version are the memory bandwidth.

A practical approach to improve the performance is to have a better memory layout and access pattern. It would help utilize the memory prefetch, reduce the memory bank conflict, and the cache line false sharing. These techniques work for both CPU and GPU.

4 Task 4

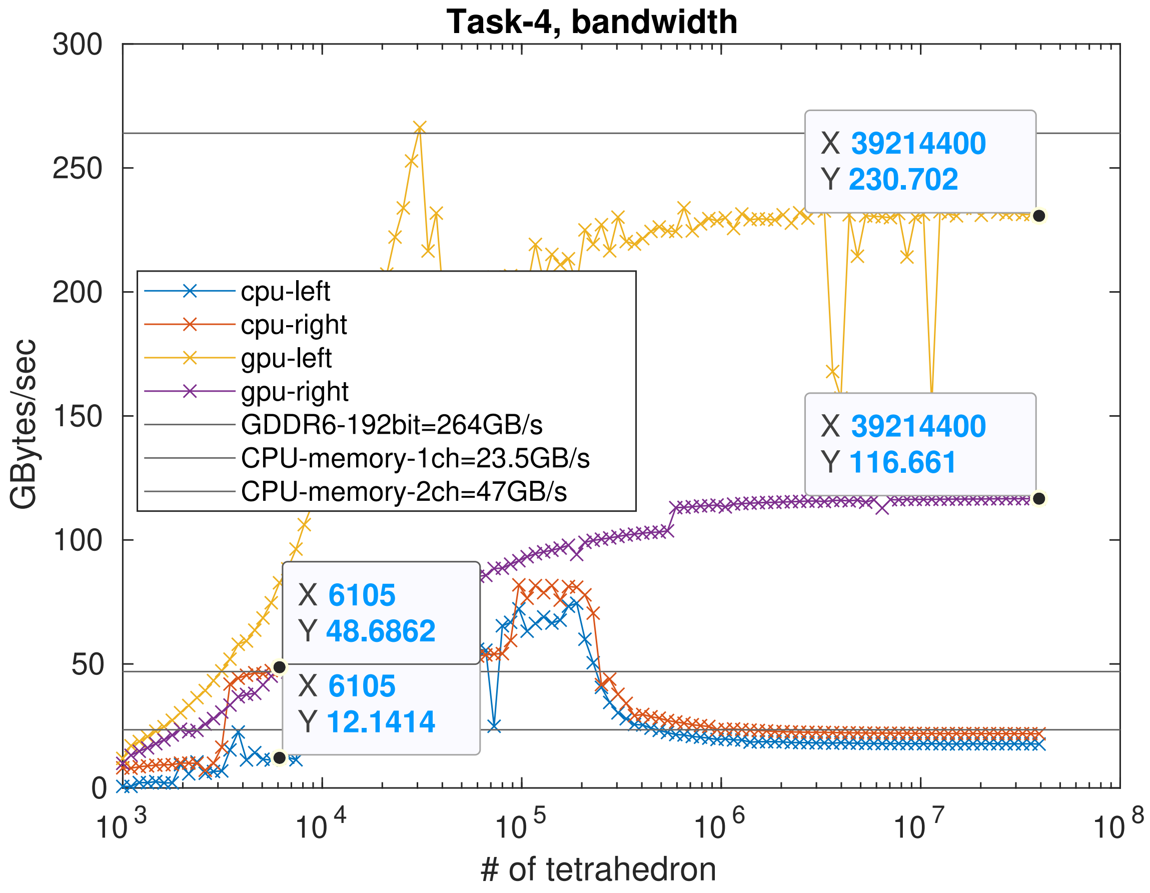

Figure 3: GPU=A3000, CPU=W-11855M, float

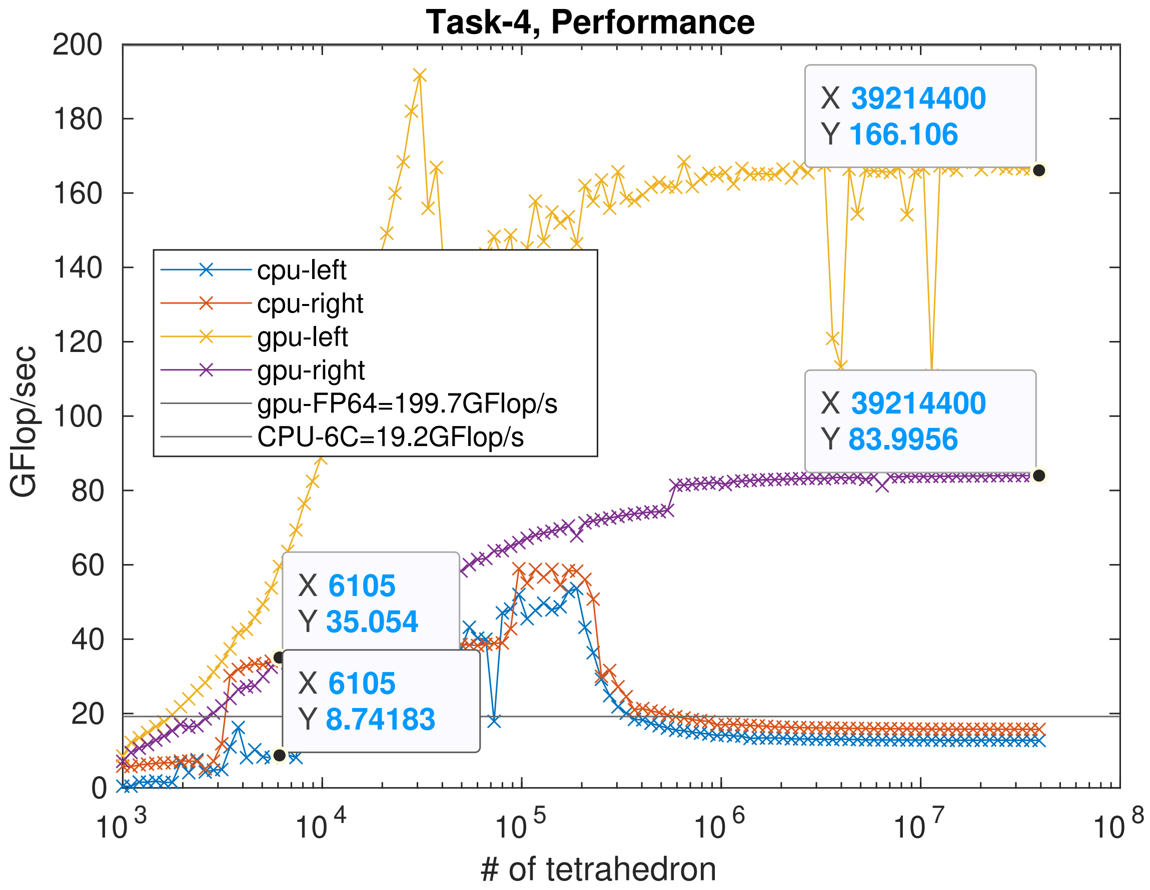

Figure 4: GPU=A3000, CPU=W-11855M, float

For the GPU, the optimal memory layout should be LayoutLeft in all memory region. The wrong memory layout could reduce the performance as much as half.

For the CPU, the optimal memory layout should be LayoutRight, especially in the cache memory region. When the matrix size is large and in the DRAM region, the performance difference between LayoutLeft and LayoutRight is relatively small. I guess the performance difference is reduced by the “random-access” feature of the memory and the CPU cache as a buffer. The performance difference in the cache memory region is large because of the “cache line false sharing” effect. It is an important issue for the OpenMP parallelization.

5 Task 5

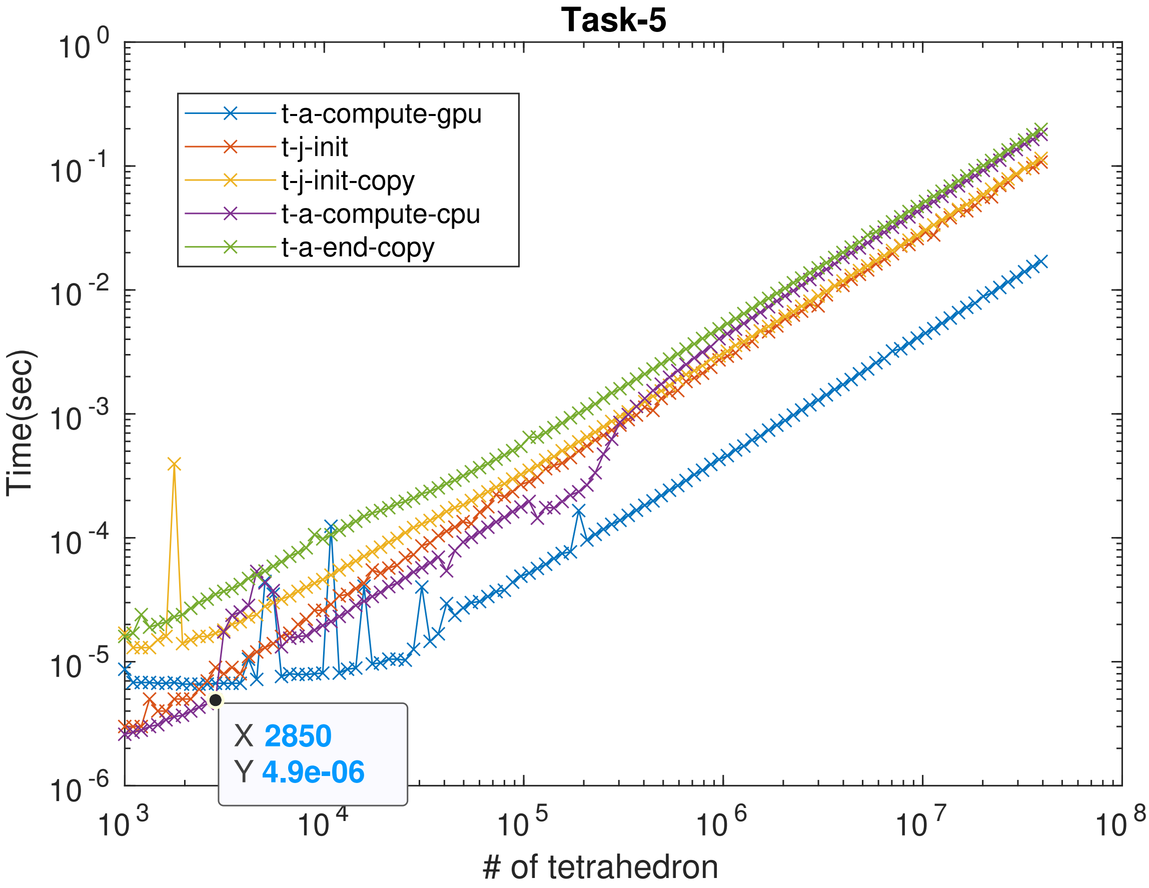

Figure 5: GPU=A3000, CPU=W-11855M, float

6 Task 6

First of all, for a specific problem size N, the program should be executed on CPU or GPU should be decided by the over all execution time, which includes the initialization, data transfer, and execution on all devices. However, the task 6 only asks to compare a specific section of the program, which does not always reflect the over all execution time.

- the Jacobians are computed from the vertices and connectivity information
  For this section, because the “t-j-init” is always smaller than the “t-j-init-copy”, it would be more efficient to calculate the J matrix on CPU than GPU.
- the element matrices are computed as above
  For this section, because the “t-a-compute-gpu” is smaller than the “t-j-compute-cpu”, it would be more efficient to calculate the A matrix on GPU than CPU. However, when the N<2850, the GPU computation is less efficient than the CPU. It means that for a small size calculation, it would be better to just use the CPU.
- the element matrices are assembled into a sparse matrix
  I do not understand this question. I cannot answer it.
- the linear system is solved with an iterative solver
  For the computation heavy section, this lab already shows that in general the GPU could have a better compute performance. Therefore, it should be executed on GPU.