DD2360 Assignment 1
Group 35
Shiyuan Shan

Exercise 1

1.1 Main differences between CPU and GPU:

CPU	GPU
A few cores for general-purpose tasks	Many smaller specialized cores for repetitive computationally intensive tasks
Allows task parallelism	Specialized for data parallelism
Handles complex control flows like branches and solves corresponding hazards	Focuses mainly on uniform control flow to lower latency in control divergence
Suits tasks for general purposes	Suits specific tasks that are repetitive and computationally intensive like rendering

1.2 Rank:

Rank	Name	GPU model
1	Frontier	AMD Instinct MI250X
2	Fugaku	
3	LUMI	AMD Instinct MI250X
4	Leonardo	NVIDIA A100
5	Summit	NVIDIA Volta GV100
6	Sierra	NVIDIA Volta GV100
7	Sunway TaihuLight	
8	Perlmutter	NVIDIA A100
9	Selene	NVIDIA A100
10	Tianhe-2A	

1.3 Power effieciency:

Rank	Name	Rmax(PFlops/s)	Power(kW)	Power efficiency (PFlop/s/kW)
1	Frontier	1194.00	22703	0.052592
2	Fugaku	442.01	29899	0.014783
3	LUMI	309.10	6016	0.05138
4	Leonardo	238.70	7404	0.032239
5	Summit	148.60	10096	0.014719
6	Sierra	94.64	7438	0.012724
7	Sunway TaihuLight	93.01	15371	0.006051
8	Perlmutter	70.87	2589	0.027374
9	Selene	63.46	2649	0.023956
10	Tianhe-2A	61.44	18482	0.003324

Exercise 2

2.1

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

Detected 1 CUDA Capable device(s)

Device 0: "Tesla T4"
  CUDA Driver Version / Runtime Version          12.0 / 11.8
  CUDA Capability Major/Minor version number:    7.5
  Total amount of global memory:                 15102 MBytes (15835398144 bytes)
  (040) Multiprocessors, (064) 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, 1638
  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 shared memory per multiprocessor:        65536 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 Managed Memory:                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 / 0 / 4
  Compute Mode:
     < Default (multiple host threads can use ::cudaSetDevice() with device simultaneously) >

deviceQuery, CUDA Driver = CUDART, CUDA Driver Version = 12.0, CUDA Runtime Version = 11.8, NumDevs = 1
Result = PASS
```

2.2 The compute capability is Version 7.5

2.3

```
[CUDA Bandwidth Test] - Starting...
Running on...

 Device 0: Tesla T4
 Quick Mode

 Host to Device Bandwidth, 1 Device(s)
 PINNED Memory Transfers
   Transfer Size (Bytes)        Bandwidth(GB/s)
   32000000                     11.9

 Device to Host Bandwidth, 1 Device(s)
 PINNED Memory Transfers
   Transfer Size (Bytes)        Bandwidth(GB/s)
   32000000                     12.9

 Device to Device Bandwidth, 1 Device(s)
 PINNED Memory Transfers
   Transfer Size (Bytes)        Bandwidth(GB/s)
   32000000                     239.5

Result = PASS

NOTE: The CUDA Samples are not meant for performance measurements. Results may vary when GPU Boost is enabled.
```

2.4 Memory bandwidth = clock rate * bus width = 5001MHz * 256bit = 160GB/s
With DDR, bandwidth gets doubled = 320GB/s
Not quite consistent with results 239.5GB/s

Exercise 3

3.1 All CUDA_FLAG related code is deleted in the makefile.

3.2 b+tree:
Cuda execution total time:

```
******command: j count=6000, rSize=6000
knodes_elem=7874, knodes_unit_mem=2068, knodes_mem=16283432
# of blocks = 6000, # of threads/block = 256 (ensure that device can handle)
Time spent in different stages of GPU_CUDA KERNEL:
 0.296627998352 s, 98.208831787109 % : GPU: SET DEVICE / DRIVER INIT
 0.000338000013 s,  0.111906453967 % : GPU MEM: ALO
 0.003873999929 s,  1.282620072365 % : GPU MEM: COPY IN
 0.000295000005 s,  0.097669824958 % : GPU: KERNEL
 0.000038999999 s,  0.012912282720 % : GPU MEM: COPY OUT
 0.000864000001 s,  0.286056727171 % : GPU MEM: FRE
Total time:
0.302038013935 s
> > > > > > > > > > > > >
 ******command: k count=10000
records_elem=1000000, records_unit_mem=4, records_mem=4000000
knodes_elem=7874, knodes_unit_mem=2068, knodes_mem=16283432
# of blocks = 10000, # of threads/block = 256 (ensure that device can handle)
Time spent in different stages of GPU_CUDA KERNEL:
 0.000021000000 s,  0.326137602329 % : GPU: SET DEVICE / DRIVER INIT
 0.000361000013 s,  5.606460571289 % : GPU MEM: ALO
 0.004612000193 s, 71.626029968262 % : GPU MEM: COPY IN
 0.000348000001 s,  5.404565811157 % : GPU: KERNEL
 0.000024000001 s,  0.372728675604 % : GPU MEM: COPY OUT
 0.001073000021 s, 16.664079666138 % : GPU MEM: FRE
Total time:
0.006438999902 s
> > > > > > > > > > > >
```

OpenMP execution total time:

```
******command: j count=6000, rSize=6000
Time spent in different stages of CPU/MCPU KERNEL:
 0.000010000000 s,  0.054389208555 % : MCPU: SET DEVICE
 0.018376000226 s, 99.945610046387 % : CPU/MCPU: KERNEL
Total time:
0.018386000767 s
> > > > > > > > > > > > >
 ******command: k count=10000
Time spent in different stages of CPU/MCPU KERNEL:
 0.000001000000 s,  0.005766015034 % : MCPU: SET DEVICE
 0.017341999337 s, 99.994239807129 % : CPU/MCPU: KERNEL
Total time:
0.017342999578 s
> > > > > > > > > > > >
```

lavaMD:

Cuda execution total time: 0.451678991318 s

OpenMP execution total time: 2.641376018524 s

3.3 b+tree: for j count = 6000 and rSize = 6000, CPU out-performs GPU, but for k count = 10000, GPU out-performs CPU.

lavaMD: GPU out-performs CPU much.

Discussion: I do observe a great performance advantage of GPU in most bench

marks, while in the rest few cases it seems to be the opposite case. This probably is due to the difference between algorithms and data sizes. GPU out-performs CPU when it comes to parallelizable computations, but it requires extra data transfer and parallelizable computation demands. Thus if the data size is not big enough or the algorithm needs sequential calculations, it is sensible that GPU performs worse than CPU does.

Exercise 4

Don’t have PDC account yet