DD2360HT22$_{\text{HW2WillenbrinkSebastian}}$

Sebastian Willenbrink

November 29, 2022

Contents

1 Repository

`https://github.com/Willenbrink/AGP/tree/master/hw_2/ex_4/rodinia_3.1`

2 Exercise 1 - Reflection on GPU-accelerated Computing

2.1 Task 1

List the main differences between GPUs and CPUs in terms of architecture.

2.1.1 CPU

- Latency-oriented architecture
- Large caches
- Out-of-order execution
- Speculative execution
- Pipelining

2.1.2 GPU

- Throughput-oriented architecture
- Simple cores
- Small caches

- Lots of ALUs
- Longer latency on a single task
- Massive tasks in parallel

2.1.3 Task 2 and 3

Check the latest Top500 list that ranks the top 500 most powerful supercomputers in the world. In the top 10, how many supercomputers use GPUs? Report the name of the supercomputers and their GPU vendor (Nvidia, AMD, ...) and model.

One main advantage of GPU is its power efficiency, which can be quantified by Performance/Power, e.g., throughput as in FLOPS per watt power consumption. Calculate the power efficiency for the top 10 supercomputers. (Hint: use the table in the first lecture)

Supercomputers with GPU accelerators: 8

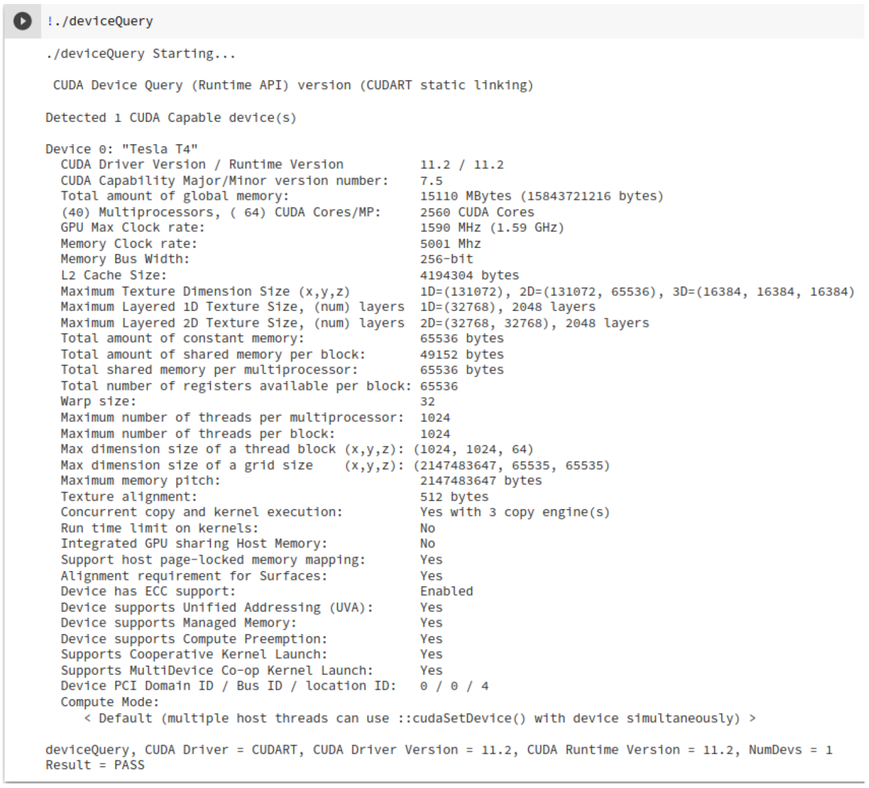
Name	Vendor	Model	PFlops/kW
Frontier	AMD	Instinct MI250X	0.052
Fugaku	N/A	N/A	0.015
LUMI	AMD	Instinct MI250X	0.051
Leonardo	Nvidia	Ampere A100	0.031
Summit	Nvidia	Tesla V100	0.015
Sierra	Nvidia	Tesla V100	0.013
Sunway TaihuLight	N/A	N/A	0.006
Perlmutter	Nvidia	Ampere A100	0.027
Selene	Nvidia	Ampere A100	0.024
Tianhe-2A	NUDT	Matrix 2000	0.003

3 Exercise 2 - Device Query

In the first assignment, you are asked to measure GPU-CPU bandwidth using the bandwidthTest utility that is included in CUDA SDK. In the same utility folder, a deviceQuery test is provided (i.e., $1_{\text{Utilities}}$/deviceQuery) to query some architectural specifications on the GPU that you are running on.

Revisit the instructions in assignment 1 on how to compile and run the utility tests either in the KTH computer room or the Google Colab platform.

3.1 The screenshot of the output from you running deviceQuery test.



```
!./deviceQuery

./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          11.2 / 11.2
  CUDA Capability Major/Minor version number:    7.5
  Total amount of global memory:                 15110 MBytes (15843721216 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 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 = 11.2, CUDA Runtime Version = 11.2, NumDevs = 1
Result = PASS
```

3.2 What architectural specifications do you find interesting and critical for performance? Please provide a brief description.

- Total amount of global memory

  Relevant to determine the maximum size of input data

- Multiprocessors, Cores/MP

  Relevant to determine execution speed

- Memory clock rate and bus width

  Relevant for transferring data, e.g. from CPU to GPU

3.3 How do you calculate the GPU memory bandwidth (in GB/s) using the output from deviceQuery? (Hint: memory bandwidth is typically determined by clock rate and bus width, and check what double date rate (DDR) may impact the bandwidth)

Presumably it is, due to DDR RAM: $clock\_rate \cdot bus\_width \cdot 2 = 320GB/s$

3.4 Compare your calculated GPU memory bandwidth with Nvidia published specification on that architecture. Are they consistent?

Yes.

4 Exercise 3 - Compare GPU Architecture

Use the Internet to search to find 3 latest Nvidia GPU architectures in which you are interested. Pick a specific model for each selected architecture, and answer the following questions:

4.1 List 3 main changes in architecture (e.g., L2 cache size, cores, memory, notable new hardware features, etc.)

Architecture	L2 Cache Size	Cores	Memory	Manufacturing process
Hopper	51200KB	16896	80GB	4nm
Ampere	40960KB	6912	80GB	7nm
Volta	6144KB	5120	32GB	12nm

4.2 List the number of SMs, the number of cores per SM, the clock frequency and calculate their theoretical peak throughput.

Architecture	Number of SMs	Cores/SM	Clock frequency	Peak Throughput
Hopper	144	128	1780MHz	2278 GB/s
Ampere	128	64	1410MHz	1805 GB/s
Volta	80	64	1530MHz	1567 GB/s

Compared where the H100, A100 80GB and the V100 32GB.

4.3 Compare (1) and (2) with the NVIDIA GPU that you are using for the course.

The Tesla T4 is inferior in many regards:

- Memory at 15GB
- Cores at 2560 Cores
- Peak throughput at 320 GB/s

It is superior in clock frequency at 5001 MHz to all above cards. These differences stem from the fact that the examined GPUs target data centers with different requirements.

5 Exercise 4 - Rodinia CUDA benchmarks and Profiling

5.1 Compile both OMP and CUDA versions of your selected benchmarks. Do you need to make any changes in Makefile?

No. Only time measurements were added and irrelevant benchmarks removed.

5.2 Ensure the same input problem is used for OMP and CUDA versions. Report and compare their execution time.

5.2.1 LU-Decomposition

Ensuring the exact same inputs was not possible as the provided example input files did not work with the CUDA version. Using a random matrix of size 1000 resulted in relatively reliable results with at most 5% deviation. OMP: 109 ms CUDA: 12 ms

5.2.2 Needle-Wunsch

Required adding a measurement for the CUDA version. For input sizes 4096, penalty 10 and 1 thread we obtain: OMP: 0.072s CUDA: 0.051s

5.2.3 BFS

Required adding a measurement to the CUDA version. For input graph1MW$_{6.txt}$ and 1 thread, we obtain: OMP: 0.066s CUDA: 0.004s

5.3 Do you observe expected speedup on GPU compared to CPU? Why or Why not?

Unfortunately, this is hard to judge. Without knowing how easy the algorithms are parallelizable I do not have any expectations regarding speedups. BFS is very simple to parallelize but what about LU-decomposition and Needle-Wunsch? Apparently Needle-Wunsch does not benefit that much whereas LU-decomposition, which I also assumed to be difficult to parallelize, benefits quite significantly.