DD2360/FDD3360 HT22 Applied GPU Programming

Assignment 2: GPU Architecture

Weichun Li weichun@kth.se
https://github.com/WCL-26/DD2360HT22

November 30, 2022

1 Reflection on GPU-accelerated Computing

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

CPUs consist of a few cores optimised for sequential serial processing, while GPUs have a massively parallel computing architecture consisting of thousands of smaller, more efficient cores (designed to handle multiple tasks simultaneously). CPUs and GPUs are very different because they are designed for two different application scenarios due to their different goals. CPUs need to be very general to handle CPUs need to be very general to handle a variety of different data types, while at the same time logical judgements introduce a lot of branch-hopping and interrupt handling. These make the internal structure of the CPU extremely complex. The GPU, on the other hand, deals with highly uniform, interdependent, large-scale data and a pure computing environment that does not need to be interrupted.

* CPUs:has more cache and local memory. CPU has a powerful ALU, which can perform arithmetic calculations in a very small number of clock cycles. Complex logic control unit. It reduces latency by providing the ability to predict branches when the program contains multiple branches.

* GPUs:has more threads, registers and SIMD units, the GPU features many ALUs and few caches (to increase memory throughput). The purpose of the cache is not to hold data that needs to be accessed later, unlike the CPU, but to serve the thread enhancement. There is also only a basic simple control structure as there is no need for branch prediction and data transfer). It also has a power-efficient and resource-rich ALU.

1.2 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.

8 of the world's 10 most powerful supercomputers use GPUs

Table 1: Supercomputers' GPUs

Name	Gpu model	GPU vendor	Power efficiency (TFlops/watts)
Frontier	Instinct 250X	AMD	52,228
Fugaku	None	None	14,783
LUMI	Instinct 250X	AMD	51,629
Leonardo	A100	NVIDIA	31,141
Summit	Tesla V100	NVIDIA	14,719
Sierra	Tesla V100	NVIDIA	12,723
Sunway TaihuLight	None	None	6,051
Perlmutter	A100	NVIDIA	27,374
Selene	A100	NVIDIA	23,983
Tianhe-2A	None	None	3,324

1.3 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)

The power efficiency of ten supercomputers has shown in Table 3

2 Device Query

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

Figure 1: screenshot of the output

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

What I find interesting is the drive and runtime version. The CUDA driver and runtime version are 11.2 and 11.2. Below is what I think is crucial to performance:

- Total amount of global memory(Data that can be stored on memory): 15110 MBytes
- CUDA Cores(number of available CUDA cores):2560 CUDA cores
- GPU max clock rate(the clock cycles per second for a core, indicating how many instructions we can run per second): 1.59 GHz
- Memory bus width(the number of bits that can be transferred):256-bit
- L2 cache size(A faster memory type for storing):4194304 bytes

2.3 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)

bandwidth = ((bus width * memory clock rate)/8) * DDR
Tesla T4's bandwidth = ((256bit * 5001MHz)/8) * 2 = 320GB/s

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

The memory bandwidth published by Nvidia for the Tesla T4 is 320 GB/s.

3 Compare GPU Architecture

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

Table 2: Three kinds of GPU's architecture

architecture	Name	Cores	Memory Size	L1 Cache	L2 Cache	Memory Clock	Memory Type
Ampere Architecture	NVIDIA RTX A6000	10752	48 GB	128 KB (per SM)	6 MB	2000 MHz	GDDR6
Turing Architecture	NVIDIA GeForce RTX 2080 SUPER	3072	8 GB	64 KB (per SM)	4 MB	1937 MHz	GDDR6
Ada Lovelace Architecture	NVIDIA GeForce RTX 4090	16384	24 GB	128 KB (per SM)	72 MB	1313 MHz	GDDR6X

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

theoretical peak throughput = Boost Clock x 2 x Cores x 109 flops

Table 3: Three kinds of GPUs' theoretical peak throughput

architecture	Name	Cores	SM	Cores per SM	Boost Clock	Theoretical Peak Throughput
Ampere Architecture	NVIDIA RTX A6000	10752	84	128	1800 MHz	38.71 TFLOPS
Turing Architecture	NVIDIA GeForce RTX 2080 SUPER	3072	48	64	1815 MHz	11.15 TFLOPS
Ada Lovelace Architecture	NVIDIA GeForce RTX 4090	16384	128	128	2520 MHz	82.58 TFLOPS

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

I am using NVIDIA Tesla T4 for the course.

- Cores: 2560
- Memory Size: 16 GB
- L1 Cache: 64 KB (per SM)
- L2 Cache: 4 MB
- Memory Type: GDDR6
- Memory Clock: 1250 MHz
- SM: 40
- Cores per SM: 64
- Boost Clock: 1590 MHz
- Theoretical Peak Throughput: 8.141 TFLOPS

4 Rodinia CUDA benchmarks and Profilings

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

I do need to make some changes in makefile to make sure some of the CUDA versions has been the correct GPU architecture.
Please see the result in the hw2_ex4.ipynb. https://github.com/WCL-26/DD2360HT22

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

Please see the result in the hw2_ex4.ipynb. https://github.com/WCL-26/DD2360HT22

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

There was a speedup with the CUDA version for both LUD and lavaMD. But for myocyte, OPM version was faster. From the result of myocyte, I think if take a long time for both GPU and CPU devices and data movement between devices when running CUDA version. So CUDA version spends longer time than openMP.