DD2360 Assignment 1

Moein Sorkhei
sorkhei@kth.se

Louiz Kim-Chan
thlkc@kth.se

3 November 2019

1 Exercise 1

1.1 Why GPUs Emerged

GPUs emerged as a suitable hardware for computer graphics, because the type of workload for the creation of images in video frame for display devices is computationally intensive. However, the computation required for each pixel is independent from other pixels, so hardware could be developed that allows for data parallel workloads without a need for synchronization or sophisticated control.

1.2 Throughput-oriented Architecture

There are two fundamental measures of processor performance, task latency and task throughput. When optimizing to run sequential programs, the goal is to minimize latency. But when optimizing to run programs where parallelism is abundant, maximizing throughput (the total amount of work completed per unit time) can result in greater performance benefit, even though the latency of single tasks is not minimized. Since tasks using the GPU, e.g. computer graphics, video processing, deep learning etc. can be done in a massively parallel manner, without the need for much control, synchronization, or irregular data accesses, a throughput-oriented architecture is preferred.

1.3 Differences between GPUs and CPUs

The main difference between CPUs and GPUs is the fact that CPUs are optimized to perform serial tasks (they have a very high clock) while GPUs are designed to break down a complex task into many smaller tasks that can be run in parallel. This, in the architecture point of view, relates to the fact that CPUs have a few large Arithmetic Logic Units (ALUs) with huge cache memory while GPUs contain many small ALUs with much smaller cahce size and can do tasks in parallel as discussed here. Hence, they could do thousands of operations at once as mentioned here.

Also, CPUs can perfrom a much wider range of tasks in comparison with GPUs because they have a larger instruction set. They are also capable of managing input and output from all the computer components. On the other hand, GPUs can do much limted range of tasks but considerably faster [1]. So, CPUs are best to perform diverse serial tasks while GPUs are best to perform a narrow range of tasks in parallel with high speed. A typical example of this advantage of GPUs is matrix multiplication. Since a lot of operations can be done in parallel when two matrices are multiplied, a GPU can be quite useful.

1.4 GPU of Lab Workstation

Stats taken from here. The lab workstation uses a GTX745, which has 3 SMs. Each SM has 128 cores. The clock frequency is 1033MHz for the GPU clock, and 900MHz (1800MHz effective) for the memory clock. The size of memory is 4GB DDR3, with 28.80GB/s bandwidth, L1 Cache of 64KB per SSM, L2 Cache of 2MB.

1.5 Scientific Paper Report

The paper *Unsupervised Scalable Representation Learning for Multivariate Time Series* [1] published in NeurIPS 2019 (one of the most popular conferences in machine learning) proposes a new deep learning method for the task of classification and regression in time series data. The authors used the PyTorch framework [2] in the Python programming language. PyTorch has both CPU and GPU support for all the popular deep learning modules. In this paper, they ran their experiments on a single Nvidia Titan Xp GPU and a single Nvidia Tesla P100 GPU with CUDA 9.0.

2 Bandwidth Test

Figure 1 shows the results of our bandwidth test across the PCIe. Both the device-host and host-device bandwidths are similar, increasing as the transfer size increases, plateauing at about 13GB/s. The increase in bandwidth is due to the tradeoff between memory allocation overhead and actual memory transfer speed. Memory allocation overhead increases with greater transfer sizes, but at low transfer sizes, the memory allocation overhead time is more significant with respect to the actual transfer time, hence a lower bandwidth. As transfer size increases, the overhead becomes less significant, hence bandwidth increases toward the maximum throughput. It likely plateaus because the maximum throughput of the PCIe 3 bus is around 16GB/s.

Figure 2 shows that device-device bandwidth is much larger than that of device-host and host-device. Bandwidth within the GPU's memory is larger than the bandwidth across the PCIe bus.

[1]https://www.maketecheasier.com/difference-between-cpu-and-gpu/

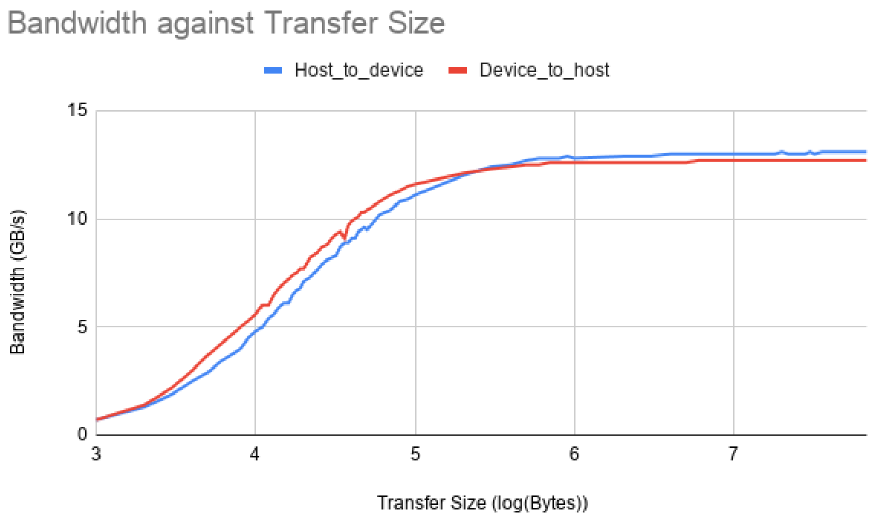

Figure 1: Bandwidth Test Results for Device-Host and Host-Device

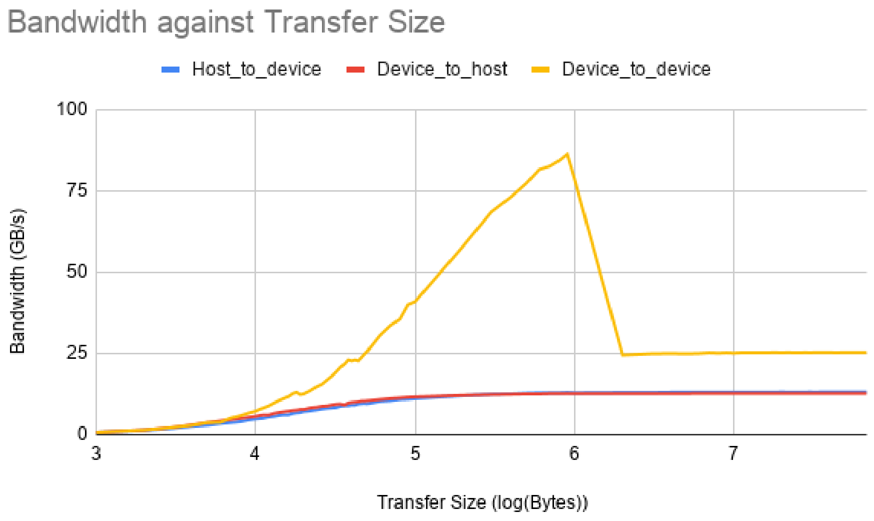

Figure 2: Bandwidth Test Results

The reason for the steep decrease in bandwidth at 900KB transfer size is likely due to caching. The size of the L2 cache in the GPU is 2MB. When transfer size is less than 1MB, it is possible for the memory transfer to be performed between memory blocks within the L2 cache, which is faster than the main GPU memory. But when the transfer size is greater than the capacity of half of the L2 cache, the memory transfer is done within the GPU's main memory, which plateaus at around 25GB/s.

References

[1] Jean-Yves Franceschi, Aymeric Dieuleveut, and Martin Jaggi. Unsupervised scalable representation learning for multivariate time series. *arXiv preprint arXiv:1901.10738*, 2019.

[2] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. 2017.