GPU - hw 1

Anirudhan J. Rajagopalan — N18824115

September 20, 2016

1 Q1

Gpu Model	Memory (GB)	Num Cores	Bandwidth (GB/sec)	Year
GTX Titan X [6]	12 GB GDDR5	3072	336.5	March, 2015
GTX 980 Ti [2]	6GB GDDR5	2816	336.5	June, 2015
GTX 1080 [5]	8 GB GDDR5X	2560	320	May, 2016
GTX 1070 [4]	8 GB GDDR5	1920	256	June, 2016
GTX 1060 [3]	6 GB GDDR5	1280	192	July, 2016

2 Q2

2.1 Bottlenecks

The main bottleneck in GPUs is its memory. While CPU's have evloved to use memory on the order of 72 GB or even 144GBs, GPUs have a maximum of 12GB so far.

2.2 Bottleneck set to continue

Based on the table above, we can see that the memory bottleneck can be expected to continue in the near future.

3 Q3

3.1 Five applications of GPU

GPUs excel in performing simple operations in parallel streams of data. Listed below are five applications of GPUs that fit this bill.

1. Computational Finance - Simulations and in Monte Carlo simulations.
2. Deep Learning & Machine Learning - Many of the deep learning problems are forumlated as matrix operations which can be done efficiently in GPUs.

3. Crypto currency mining - Since crypto currency mining is a repeated calculation of SHA256 checksums [7]

4. Graphics rendering - Graphics and computer animations are inherently small vector operations done at multiple pixel positions.

5. Defense and Intelligence [1] - Defense and intelligence requires data from wide array of sensors which has to be collected parallely and converted into actionable items in real time. GPUs help in this process.

4 Q4

4.1 Impact of the amount of data on GPU performance

Yes, The amount of data has a definite influence on the performance of the system. If the amount of data is small, then moving the information from CPU's memory to GPU will be a big component of the time taken to perform the task and due to this, we might not observe any definite performance gains. But if the amount of data is huge, using GPUs will most certainly give better performance in this scenario.

5 Q5

5.1 Beneficial for GPU implementation?

5.1.1 Finding whether a number exists in an array of 10M numbers

This can essentially be done as performing the simple equals operation on 10M numbers at the same time. Since this involves a simple operation on independent data, we can implement this in a GPU.

5.1.2 Calculating the first 1M fibonacci numbers

Fibonacci number calculation is essentially a sequential operation which is difficult to parallelize. So it is not well suited for GPU implemntation.

5.1.3 Multiplying 1000×1000 matrices

Matrix multiplication is well suited for GPU implementation. There are parallel algorithms for matrix multiplication which can then be implemented using GPUs.

References

[1] Nvidia. Defense and Intelligence. `http://www.nvidia.com/object/tesla-defense-intelligence.html`.

[2] Nvidia. Geforce 980 Ti. `http://www.geforce.com/hardware/desktop-gpus/geforce-gtx-980-ti/speci`

[3] Nvidia. Geforce GTX 1060. http://www.geforce.com/hardware/10series/geforce-gtx-1060.

[4] Nvidia. Geforce GTX 1070. http://www.geforce.com/hardware/10series/geforce-gtx-1070.

[5] Nvidia. Geforce GTX 1080. http://www.geforce.com/hardware/10series/geforce-gtx-1080.

[6] Nvidia. Geforce Titan X. http://www.geforce.com/hardware/desktop-gpus/geforce-gtx-titan-x/sp

[7] David Perry. Why is a GPU better tool for mining? http://bitcoin.stackexchange.com/a/16505.