**GPU and TPU**

*Brief by Pulkit Kalia, Kajal Dalvi*

***Why should you care?*** – **GPU** and **TPU** are composed of hundreds of cores that can handle thousands of threads simultaneously which results in extremely efficient **mathematical computations** which forms the basis for many big data tasks. The strength of GPU lies in data parallelization and memory bandwidth. TPU (optimized for TensorFlow) can handle the massive multiplications and additions for neural networks, at fast speeds with less power consumption.

***Approach –***. GPU comprises of several cores where each of these cores have multiple functional units, such as arithmetic and logic units. One or more of these functional units are used to process each thread of execution and a group of such units are called **thread processors**. All thread processors in a core of GPU perform the same instructions, as they share the same control unit which helps to perform the **same instruction** on data in parallel.[[Ref](https://www.kdnuggets.com/2016/04/basics-gpu-computing-data-scientists.html)]

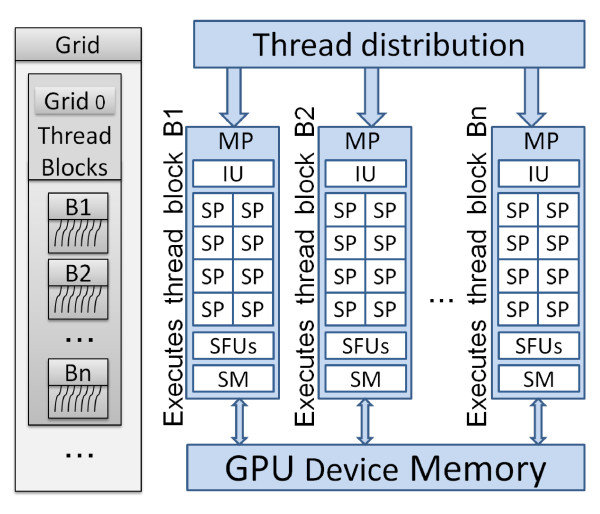


Figure 1: Simplified GPU Architecture: MP=Multi Processor, SM=Shared Memory, SFU=Special Functions Unit, IU=Instruction Unit, SP=Streaming processor (core). ([Chikkagoudar, Satish & Wang, Kai & Li, Mingyao](https://www.researchgate.net/publication/51168475_GENIE_A_software_package_for_gene-gene_interaction_analysis_in_genetic_association_studies_using_multiple_GPU_or_CPU_cores). (2011).)

The TPU includes the following computational resources:

* Matrix Multiplier Unit (MXU): 65,536 8-bit multiply-and-add units for matrix operations
* Unified Buffer (UB): 24MB of SRAM that work as registers
* Activation Unit (AU): Hardwired activation functions

The MXU, UB and AU proceed with operations which are defined using dozens of high-level instructions specifically designed for neural network inference.[[Ref](https://cloud.google.com/blog/products/gcp/an-in-depth-look-at-googles-first-tensor-processing-unit-tpu)]

***Results –*** Inefficiency of CPUs to process big data led to GPUs being the gold standard in compute-heavy tasks and has shown remarkable performance over CPUs. However, GPUs are very expensive. TPUs are a lot cheaper and easily available on the cloud and exhibit immense improvements in speed and performance. The following table details the key differences between them.

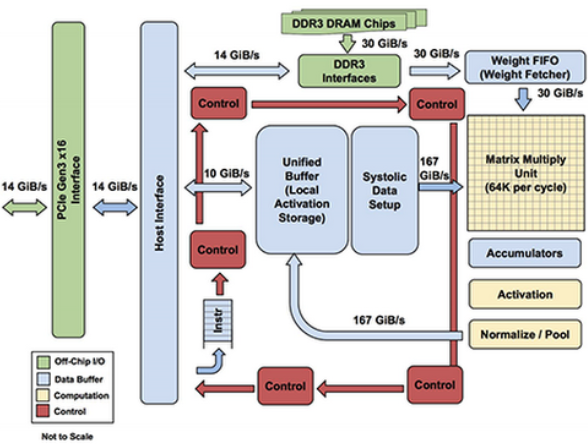


Figure 2: TPU block diagram. **(**[Google Cloud Documentation](https://cloud.google.com/blog/products/gcp/an-in-depth-look-at-googles-first-tensor-processing-unit-tpu)).

|  |  |  |  |
| --- | --- | --- | --- |
|  | **CPU** | **GPU** | **TPU** |
| Speed | Low | Moderate | High |
| Cost | Expensive at $3191 per training preemption on CPU | Expensive at $3454 per training on an 8- GPU Tesla V100 | Cost-Efficient at $724 per training preemption |
| Best suited for | Achieves the highest FLOPS utilization for RNNs and supports the largest model because of large memory capacity | Shows better flexibility and programmability for irregular computations, such as small batches and non MatMul computations | Highly optimized for large batches and CNNs and has the highest training throughput |

Table 1: All the experiments were run on a Google Compute n1-standard-2 machine with 2 CPU cores and 7.5GB of memory.

***Pros/Cons***

**GPU:**

* Pro: Exhibits great speed and performance for mathematical computations and parallel tasks. Shows good flexibility and support for different programming frameworks.
* Con: GPU needs CPU to run the instructions. GPUs performance is highly dependent on the quality of code. They are also expensive.

**TPU:**

* Pro: Has the highest training throughput. They are more power and cost efficient.
* Con: They are specialized to work with TensorFlow only and currently cannot be used with other frameworks.

***Conclusion –*** GPU and TPU have made big data tasks faster by leveraging parallel execution. Both are widely used these days in deep learning and have become a go-to hardware for such applications.