A
Seminar Report
On

"The Performance Analysis & an In-depth Look at Google's Tensor Processing Unit"

For the degree of

MASTER OF TECHNOLOGY
(Computer Science and Engineering)

Submitted by

Naik Durgesh Sunil
(2018MTECSCO005)

Under the guidance of
Prof. N. K. Pikle

Department of Computer Science and Engineering

WALCHAND COLLEGE OF ENGINEERING, SANGLI
2018-2019

CERTIFICATE

This is to certify that the dissertation work entitled "The Performance Analysis & an In-depth Look at Google's Tensor Processing Unit" submitted by "*Naik Durgesh Sunil (2018MTECSCO005)*" in partial fulfilment of the requirement for the degree of "*Master of Technology (Computer Science and Engineering*)" is a record of his/her own work carried out under my supervision during the year *2018-19*.

Date:

Prof. Dr. B. F. Momin
(Head of Department)

Prof. N. K. Pikle
(Seminar Guide)

Prof. M. A. Shah
(Panel Member)

Prof S. S. Solapure
(Panel Member)

ACKNOWLEDGMENT

I would like to express my deep gratitude towards my seminar Guide Prof. N. K. Pikle for his constant guidance during my seminar work and to the Computer Science and Engineering Department for making its facilities available to me. I take this opportunity to express my sincere thanks to all the staff members of Computer Science and Engineering Department for their help whenever required. Finally, I express my sincere thanks to all those who helped me directly or indirectly while executing seminar work.

Naik Durgesh Sunil
(2018MTECSCO005)

CONTENTS

LIST OF FIGURES

LIST OF TABLES

1. ABSTRACT

Many architects believe that major enhancements in cost-energy performance must now come from domain-specific hardware. The paper evaluates a custom ASIC called a Tensor processing Unit (TPU) deployed in datacenters since 2015 that accelerates the inference part of neural networks (NN). The heart of the TPU is a 65,536 8-bit mac matrix multiply unit that provides a peak throughput of 92 TeraOps/second (TOPS) and a large (28 MiB) software-managed on-chip memory. The TPU's deterministic execution model is a better match to the 99th-percentile response time demand of their NN applications than the time-varying optimizations of CPUs and GPUs that help average throughput over secure latency. The shortage of such options helps justify why, despite having myriad MACs and a big memory, the TPU is comparatively little and low power. It's tends to compare the TPU to a server-class CPU and a Nvidia GPU, that area unit contemporaries deployed within the same datacenters. The workload, written in the high-level TensorFlow framework, uses production NN applications (MLPs, CNNs, and LSTMs) that represent 95th of Google's datacenters NN inference demand. Despite low utilization for a few applications, the TPU is on average about 15-30 times quicker than its modern GPU or CPU, with TOPS/Watt about 30-80 times higher. Moreover, using the GPU's GDDR5 memory in the TPU would triple achieved tops and raise TOPS/Watt to nearly 70 times the GPU and 200 times the CPU.

Google began searching for a way to support neural networking for the development of their services such as voice recognition but proposing to development of a new architecture instead of using existing hardware, they would require twice as many data centers, that's why Norman Jouppi begins work on a new architecture to support TensorFlow to achieve high performance in complex computation.

Keywords: ASIC, TPU, GPU, TOPS, LSTM.

1.1 Literature Survey:

The tensor processing unit was announced in May 2016 at Google I/O, when the company said that the TPU had already been used inside their data centers for over a year. The chip has been specifically designed for Google's TensorFlow framework, a symbolic math library which is used for machine learning application such as neural network. However, Google still uses CPUs and GPUs for other types of machine learning. Other AI accelerator designs are appearing from other vendors also are aimed at embedded and robotics markets.

Google's TPUs are proprietary and are not commercially available, although on February 12, 2018, The New York Times reported that Google "would allow other companies to buy access to those chips through its cloud-computing service." Google has stated that they were used in the AlphaGo versus Lee Sedol series of man-machine Go games, as well as in the AlphaZero system which produced Chess, Shogi and Go playing programs from the game rules alone and went on to beat the leading programs in those games. Google has also used TPUs for Google Street View text processing, and was able to find all the text in the Street View database in less than five days. In Google Photos, an individual TPU can process over 100 million photos a day. It is also used in RankBrain which Google uses to provide search results.

1.1.1 First Generation TPU

The first generation TPU is an 8-bit matrix multiplication engine, driven with CISC instruction by the host processor across a PCIe 3.0 bus. It manufactured on a 28 mm process with a die size ≤ 331 mm^2. The clock speed is 700MHz and it has a thermal design power of 28-40W. It has 28 MiB of on chip memory, and 4 MiB of 32-bit accumulator taking the result of 256x256 systolic array of 8-bit multiplier. Within the TPU package is 8GiB of dual-channel 2133 MHz DDR3 SDRAM offering 34 GB/s of bandwidth. Instruction transfer data to or from the host, perform matrix multiplications or convolutions, and apply activation function.

1.1.2 Second Generation TPU

The second-generation TPU was announced in May 2017. Google stated the first-generation TPU design was limited by memory bandwidth and using 16 GB of High Bandwidth Memory in the second-generation design increased bandwidth to 600GB/s and performance to 45 TFLOPS. The TPUs are then arranged into four-chip modules with a performance of 180 TFLOPS. Then 64 of these modules are assembled into 256-chip pods with 11.5 PFLOPS of performance. Notably, while the first-generation TPUs were limited to integers, the second-generation TPUs can also calculate in floating point. This makes the second-generation TPUs useful for both training and inference of machine learning models. Google has stated these second-generation TPUs will be available on the Google Compute Engine for use in TensorFlow applications.

1.1.3. Third Generation TPU

The third-generation TPU was announced on May 8, 2018. Google announced that processors themselves are twice as powerful as the second-generation TPUs, and would be deployed in pods with four times as many chips as the preceding generation. This results in an 8-fold increase in performance per pod (with up to 1,024 chips per pod) compared to the second-generation TPU deployment.

1.1.4. Edge TPU

In July 2018 the Edge TPU was announced. Edge TPU is Google's purpose-built ASIC chip designed to run machine learning (ML) models for edge computing. It uses the Cloud IoT Edge software stack, which combines gateway functions using the Edge IoT Core software, with Edge ML, a machine learning runtime based on TensorFlow Lite.

Before going with detail architecture of TPU, it is necessary to having idea about some concept which is related to TPU.

2. INTRODUCTION

The raising markets of AI-based data analytics and deep learning applications, such as software for self-driving cars, have pushed several companies to develop specialized hardware to boost the performance of large dense matrix (tensor) computation. This is essential to both training and inferencing of deep learning applications. For instance, Google designed the Tensor Processing Unit specifically for tensor calculations. Recently, NVIDIA released the Volta micro architecture featuring specialized computing units called Tensor Cores. An NVIDIA Tensor Core is capable of performing one matrix-multiply-and-accumulate operation on a 4×4 matrix in one GPU clock cycle. In mixed-precision mode, Tensor Cores take input data in half floating-point precision, perform matrix multiplication in half precision and the accumulation in single precision. The NVIDIA Tesla V100 GPU provides a total of 640 Tensor Cores that can reach a theoretical peak performance of 125 T flops/s. Hence, systems like the NVIDIA DGX-1 system that combines eight Tesla V100 GPUs could achieve a theoretical peak performance of one Pflops/s in mixed precision. The pre-exa scale systems, such as the Summit supercomputerthathassixTeslaV100GPUs connected with high-speed NVLink in each compute node for a total of 4,600 nodes, will offer nearly 18M Tensor Cores!

2.1. DEEP NEURAL NETWORKS

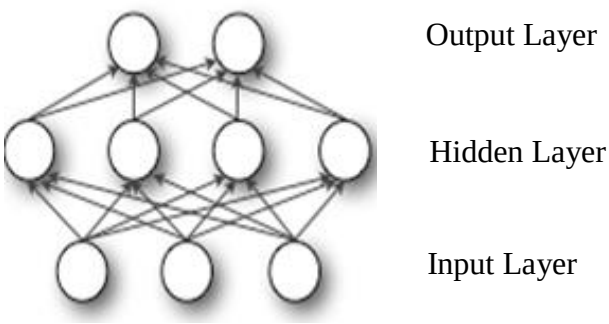

Figure 1: DNN

In deep-learning networks, each layer of nodes trains on a distinct set of features based on the previous layer's output.

There are two phases of DNN:

1.Training (Learning)

2. Inference (Prediction)

The two phases of NN are called training (or learning) and inference (or prediction), and they refer to development versus production. The developer chooses the number of layers and the type of NN, and training determines the weights. Virtually all training today is in floating point, which is one reason GPUs have been so popular. A step called quantization transforms floating point numbers into narrow integers often just 8 bits which are usually good enough for inference. Eight-bit integer multiplies can be 6X less energy and 6X less area than IEEE 754 16-bit floating-point multiplies, and the advantage for integer addition is 13X in energy and 38X in area

Three kinds of NNs are popular today:

1. Multi-Layer Perceptrons (MLP): Each new layer is a set of nonlinear functions of a weighted sum of all outputs (fully connected) from the prior one.

2. Convolutional Neural Networks (CNN): Each layer is a set of nonlinear functions of weighted sums at different coordinates of spatially nearby subsets of outputs from the prior layer, which allows the weights to be reused.

3. Recurrent Neural Networks (RNN): Each subsequent layer is a collection of nonlinear functions of weighted sums of outputs and the previous state. The most popular RNN is Long Short-Term Memory (LSTM). The art of the LSTM is in deciding what to forget and what to pass on as state to the next layer. The weights are reused across time steps.

2.2. ARCHITECTURE OF TENSOR PROCESSING UNIT (TPU)

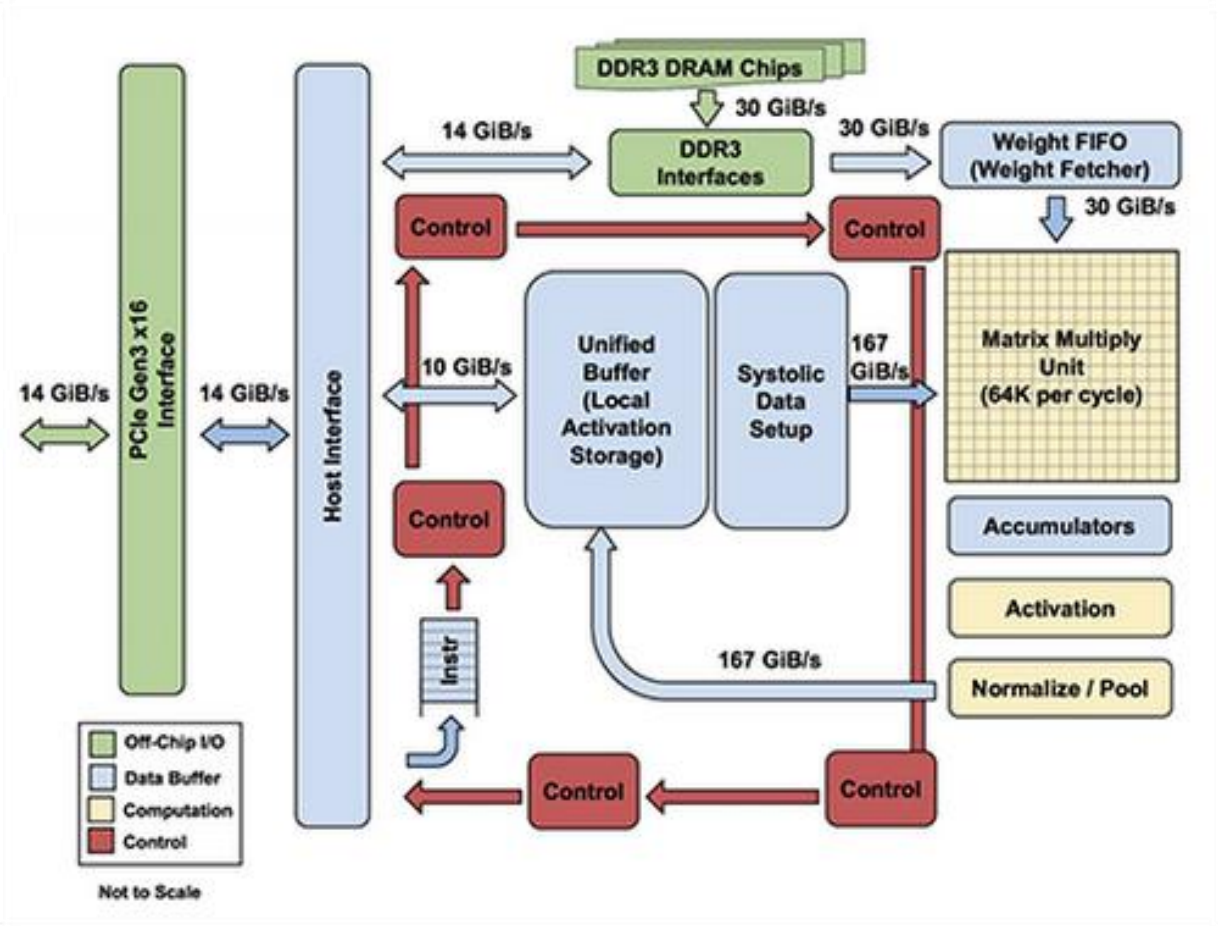

Figure 2: Block Diagram of TPU

Starting as early as 2006, Google deploying GPUs, FPGAs, or custom ASICs in their datacenters. They concluded that the few applications that could run on special hardware could be done virtually for free using the excess capacity of their large datacenters, and it's hard to improve on free. That changed in 2013 when a projection showed people searching by voice for three minutes a day using speech recognition DNNs would double their datacenters' computation demands, which would be very expensive using conventional CPUs. Thus, they started a high priority project to produce a custom ASIC quickly for inference (and bought off-the-shelf GPUs for training). The goal was to improve cost-performance by 10X over GPUs.

Rather than be tightly integrated with a CPU, to reduce the probabilities of delaying deployment, the TPU was designed to be a coprocessor on the PCIe I/O bus, permitting it to plug into existing servers just as a GPU does. Moreover, to simplify hardware design and debugging, the host server sends TPU instructions for it to execute rather than the TPU fetching them itself. Hence, the TPU is nearer in spirit to an FPU (floating-point unit) coprocessor than it's to a GPU.

The goal was to run whole inference models in the TPU to reduce interactions with the host CPU and to be flexible enough to match the NN needs of 2015 and beyond, instead of just what was required for 2013 NNs.

The TPU instructions are sent from the host over the PCIe Gen3 x16 bus into an instruction buffer. The internal blocks are typically connected together by 256-byte-wide paths. Starting in the upper-right corner, the Matrix Multiply Unit is the heart of the TPU. It contains 256x256 MACs that can perform 8-bit multiply and-adds on signed or unsigned integers.

The philosophy of the TPU microarchitecture is to keep the large matrix unit busy. Toward that end, the instruction that reads the weights follows the decoupled-access/execute philosophy,4 in that it can complete after sending its address, but before the weight is fetched from weight memory. The matrix unit will stall if the input activation or weight data is not ready.

As reading a large SRAM uses much more power than arithmetic, the matrix unit uses systolic execution to save energy by reducing reads and writes of the unified buffer. It relies on data from different directions arriving at cells in an array regularly where they are combined. A given 256-element matrix-vector multiplication moves through the matrix as a diagonal wave front. The weights are preloaded and take effect with the advancing wave alongside the first data of a new block. Control and data are pipelined to give software the illusion that the matrix unit reads 256 inputs simultaneously, and that they instantly update one location of each of the 256 accumulators.

The goal was to run whole inference models in the TPU to reduce I/O between the TPU and the host CPU. Moreover, to simplify hardware design and debugging, the host server sends TPU instructions over the PCIe bus for it to execute rather than having the TPU sequence itself. Hence, the TPU is closer in spirit to a floating-point coprocessor than it is to a GPU. As instructions travel over this relatively slow bus, the TPU follows the CISC tradition. It has about a dozen instructions, such as Read_Host_Memory, Read_Weights, Matrix_Multiply, and Write_Host_Memory. The average clock cycles per instruction of these CISC instructions is typically 10 to 20. Following table shows the instruction with their description.

Table1 1: TPU Instruction and Function

TPU Instruction	Function
Read_Host_Memory:	it reads data from the CPU host memory into the Unified Buffer (UB).
Read_Weights:	it reads weights from Weight Memory into the Weight FIFO as input to the Matrix Unit
MatrixMultiply /Convolve:	it multiplies and convolve with data and weights accumulate the result.
Activate:	it performs the nonlinear function of the artificial neuron, with options for Sigmoid, and so on.
Write_Host_Memory:	it writes data from the Unified Buffer into the CPU host memory.

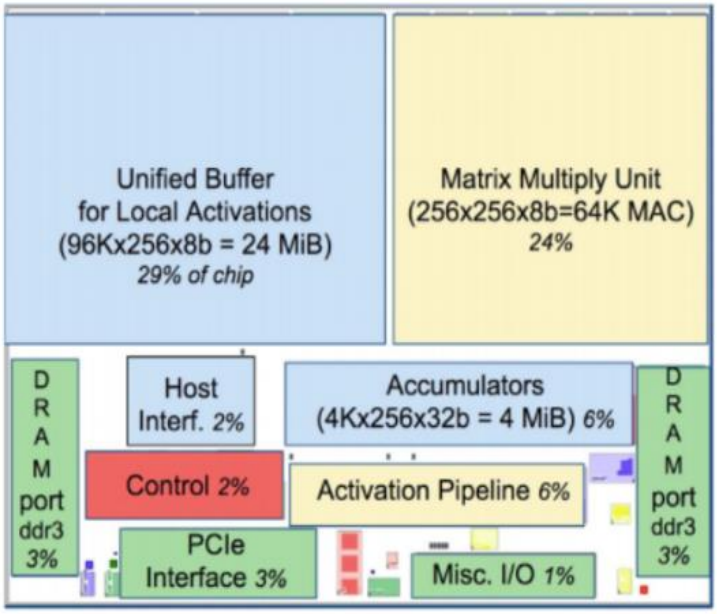

Figure 3: Floorplan of TPU die

Figure 3 shows the floor plan of the TPU die. The 24 MiB Unified Buffer is almost a third of the die and the Matrix Multiply Unit is a quarter, so the data path is nearly two-thirds of the die. The 24 MiB size was picked in part to match the pitch of the Matrix Unit on the die and, given the short development schedule, in part to simplify the compiler. Control is just 2%. Figure 3 shows the TPU on its printed circuit card, which inserts into existing servers like a SATA disk.

The MXU (Matrix Multiply Unit) is heart of TPU and goal is to keep it busy always in computation.

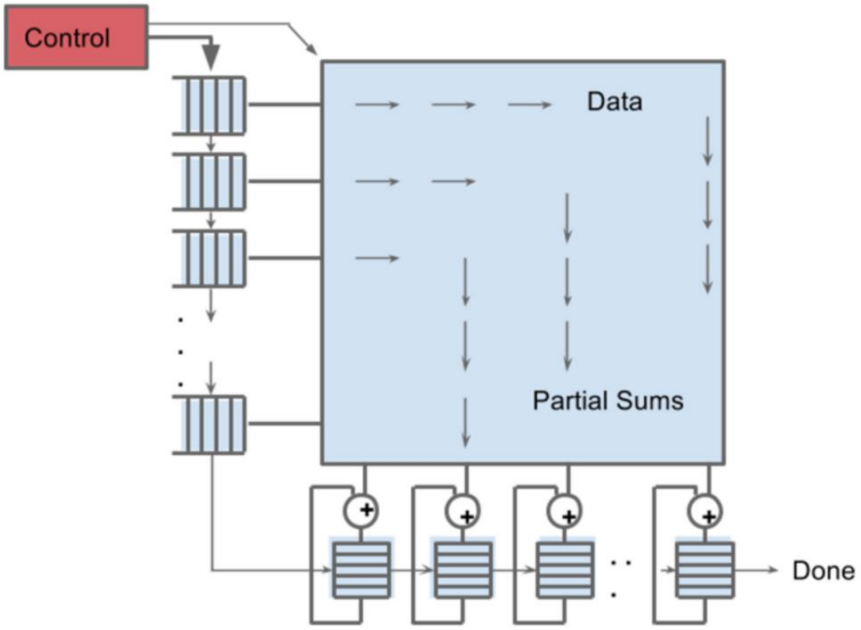

Figure 4: Matrix Multiply Unit

TPU Matrix Multiply Unit has a systolic array mechanism that contains 256x256 = 65,536 ALU's. That means TPU can process 65,536 multiply-and-add for 8bit integers every cycle. TPU runs at 700 MHz, it can compute 92 Teraops per second in Matrix Unit.

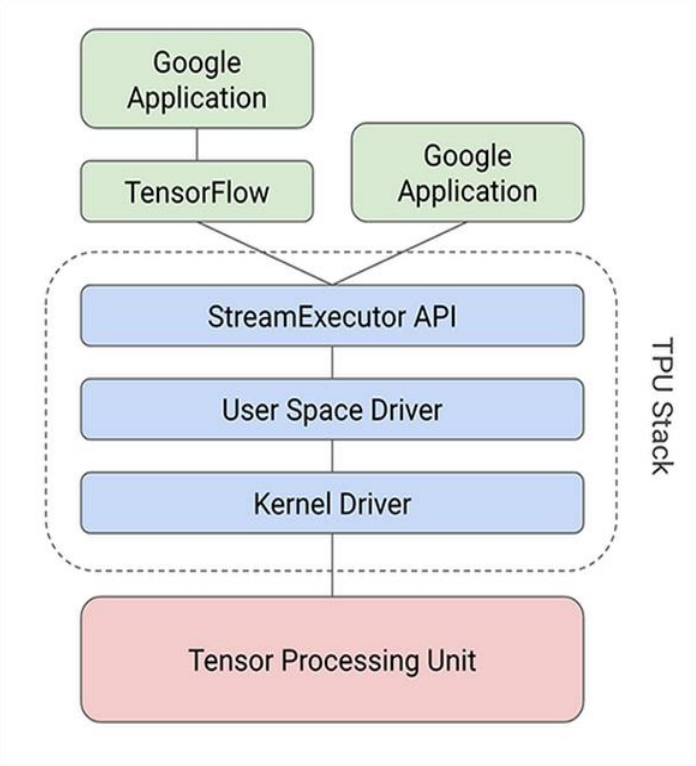

Figure 5: TPU Stack

The TPU Stack is layered architecture in which TPU stack translate the API calls into the TPU instruction. First TPU accepts the instruction through some Google application which uses the TensorFlow library specially for GPU and TPU or from some simple application design by user. The User Space driver changes frequently. It sets up and controls TPU execution, reformats data into TPU order, translates API call into TPU instruction, and turns into them application binary. The User Space driver compiles a model the first time it is evaluated, caching the program image and writing the weight image into the TPU's weight memory; the second and following evaluations run at full speed. The Kernel Driver is lightweight and handles only memory management and interrupts.

2.3. Implementation: Environment

Table 2: Environmental Setup.

CPU	**GPU**	**TPU**
Intel Xeon CPU	Nvidia Tesla K80	Google Edge TPU
CPU Core → 1 (Hyperthreaded)	CUDA Cores → 2496	TPU Cores → 2
CPU Clock → 2.20GHz	GPU base clock → 875MHz	TPU Clock → 700MHz
L3 Cache → 45 MB	Memory → 12,288 MB GDDR5	Weight Memory → 8,192 MB DRAM

The Computing platform used in experiments is a Google Colab that is cloud based computing with Python programming environment and Linux based operating system. The platform includes a high-performance Tensor Processing Unit (TPU) having 2 single threaded cores each has 700 MHz clock frequency with 8 GB of DRAM for weight memory.

For comparison-based evaluation, the implementation platform also includes high-performance Nvidia Tesla K80 GPU and Intel's Xeon CPU with hyper threaded single CPU core having 2.20 GHz clock frequency in which built in 45 MB of L3 cache. High speed shared memory and synchronization mechanism are adapted in SMs of GPU. There are 20 SMs in GPU, each of which has 4 SPs. The SP's clock frequency is 0.875 GHz. In each block, there are 65,536 available registers. Each multiprocessor has 128 CUDA cores.

For testing the performance of TPU Dataset of 60,000 28x28 grayscale images of 10 fashion categories, along with a test set of 10,000 images.

Table 3: Labels of Test set images.

Label	**Description**
0	T-shirt/top
1	Trouser
2	Pullover
3	Dress
4	Coat
5	Sandal
6	Shirt
7	Sneaker
8	Bag
9	Ankle boot

The programs which are used for demonstration purpose, contains training set of 60,000 labeled images which having 0-9 labels as shown in above Table No.3. The testing set contains 10,000 of unlabeled images which are to be classified.

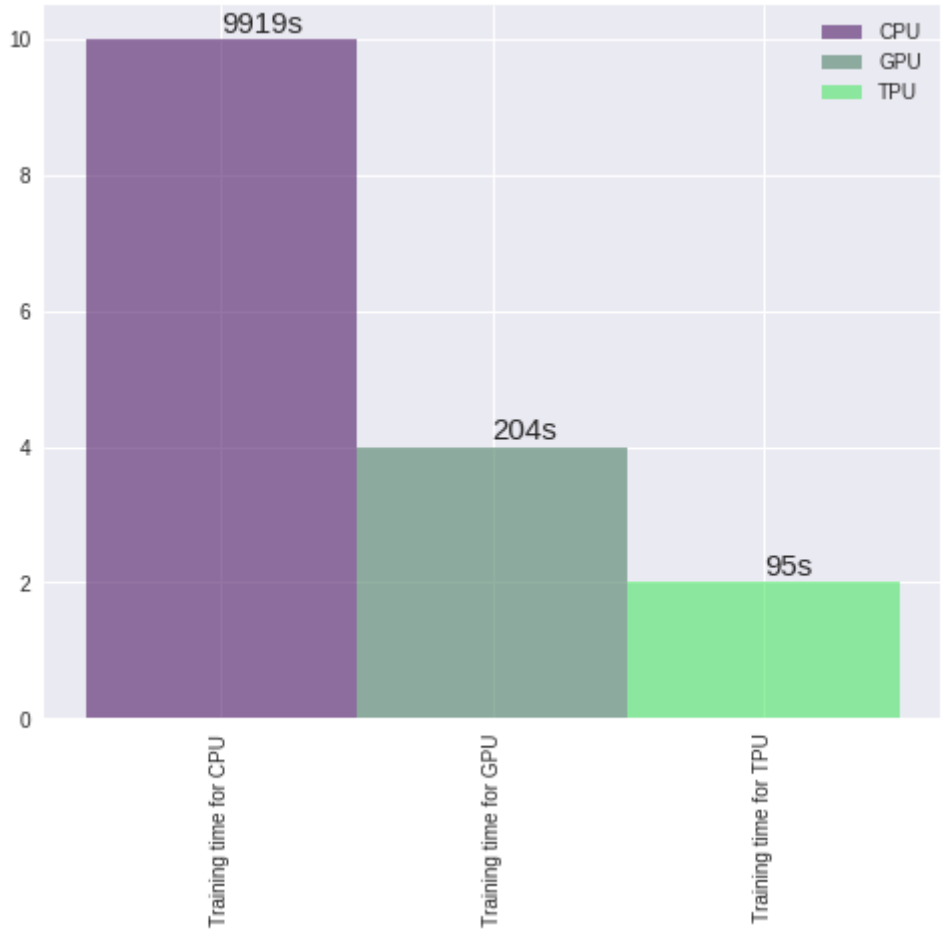

Figure 6: Performance Plot of CPU, GPU, TPU

The average training time taken by CPU is 9919 seconds and the average training time taken to run the same program on GPU is 204 seconds, which got 48-time speedup than CPU. The average training time taken to run the same program on TPU is 95 seconds, which got 104 times speedup than CPU and 3 times speedup than GPU.

2.4. Use Of TPU:

2.4.1. RankBrain algorithm used by Google search:

RankBrain is a machine learning-based search engine algorithm, the use of which was confirmed by Google on 26 October 2015. It helps Google to process search results and provide more relevant search results for users. In a 2015 interview, Google commented that RankBrain was the third most important factor in the ranking algorithm along with links and content. As of 2015, "RankBrain was used for less than 15% of queries." The results show that RankBrain produces results that are well within 10% of the Google search engine engineer team.

2.4.2. Google Photos:

Google Photos gives users free, unlimited storage for photos up to 16 megapixels and videos up to 1080p resolution. The service automatically analyses photos, identifying various visual features and subjects. Users can search for anything in photos, with the service returning results from three major categories: People, Places, and Things. Google Photos recognizes faces, grouping similar ones together geographic landmarks (such as the Eiffel Tower); and subject matter, including birthdays, buildings, animals, food, and more.

Different forms of machine learning in the Photos service allow recognition of photo contents, automatically generate albums, animate similar photos into quick videos, surface past memories at significant times, and improve the quality of photos and videos. In May 2017, Google announced several updates to Google Photos, including reminders for and suggested sharing of photos, shared photo libraries between two users, and physical albums, with Photos automatically suggesting collections based on face, location, trip, or other distinction.

2.4.3. Google Translate:

Google Translate is a free multilingual machine translation service developed by Google, to translate text. It offers a website interface, mobile apps for Android and iOS, and an API that helps developers build browser extensions and software applications. Google Translate supports over 100 languages at various levels.

During a translation, it looks for patterns in millions of documents to help decide on the best translation. Its accuracy has been criticized and ridiculed on several occasions. In November 2016, Google announced that Google Translate would switch to a neural machine translation engine - Google Neural Machine Translation (GNMT) which translates "whole sentences at a time, rather than just piece by piece. It uses this broader context to help it figure out the most relevant translation, which it then rearranges and adjusts to be more like a human speaking with proper grammar". Originally only enabled for a few languages in 2016, GNMT is gradually being used for more languages.

2.4.4. Google Cloud Platform:

Google Cloud Platform (GCP), offered by Google, is a suite of cloud computing services that runs on the same infrastructure that Google uses internally for its end-user products, such as Google Search and YouTube. Alongside a set of management tools, it provides a series of modular cloud services including computing, data storage, data analytics and machine learning. Registration requires a credit card or bank account details. Google Cloud Platform provides Infrastructure as a service, Platform as a service, and Serverless computing environment.

In April 2008, Google announced App Engine, a platform for developing and hosting web applications in Google-managed data centers, which was the first cloud computing service from the company. The service became generally available in November 2011. Since the announcement of App Engine, Google added multiple cloud services to the platform.

Google Cloud Platform is a part of Google Cloud, which includes the Google cloud platform public cloud infrastructure, as well as G Suit, enterprise versions of Android and Chrome OS, and API's for machine learning and enterprise mapping services.

Cloud TPUs are optimized for specific workloads. In some situations, you might want to use GPUs or CPUs on Compute Engine instances to run your machine learning workloads. In general, you can decide what hardware is best for your workload based on the following guidelines:

2.4.4.1. CPUs

- Quick prototyping that requires maximum flexibility
- Simple models that do not take long to train
- Small models with small effective batch sizes
- Models that are dominated by custom TensorFlow operations written in C++
- Models that are limited by available I/O or the networking bandwidth of the host system

2.4.4.2. GPUs

- Models that are not written in TensorFlow or cannot be written in TensorFlow
- Models for which source does not exist or is too onerous to change
- Models with a significant number of custom TensorFlow operations that must run at least partially on CPUs
- Models with TensorFlow ops that are not available on Cloud TPU (see the list of available TensorFlow ops)
- Medium-to-large models with larger effective batch sizes

2.4.4.3. TPUs

- Models dominated by matrix computations.
- Models with no custom TensorFlow operations inside the main training loop.
- Models that train for weeks or months.
- Larger and very large models with very large effective batch sizes.

Cloud TPUs are **not** suited to the following workloads:

- Linear algebra programs that require frequent branching or are dominated element-wise by algebra. TPUs are optimized to perform fast, bulky matrix multiplication, so a workload that is not dominated by matrix multiplication is unlikely to perform well on TPUs compared to other platforms.
- Workloads that access memory in a sparse manner might not be available on TPUs.
- Workloads that require high-precision arithmetic. For example, double-precision arithmetic is not suitable for TPUs.
- Neural network workloads that contain custom TensorFlow operations written in C++. Specifically, custom operations in the body of the main training loop are not suitable for TPUs.

Neural network workloads must be able run multiple iterations of the entire training loop on the TPU. Although this is not a fundamental requirement of TPUs themselves, this is one of the current constraints of the TPU software ecosystem that is required for efficiency.

3. Future Development

- Allows to build machine learning supercomputers called “TPU Pods”.
- Improvement in training times.
- Allows mixing and matching with other hardware which includes Skylake CPUs and NVIDIA GPUs.

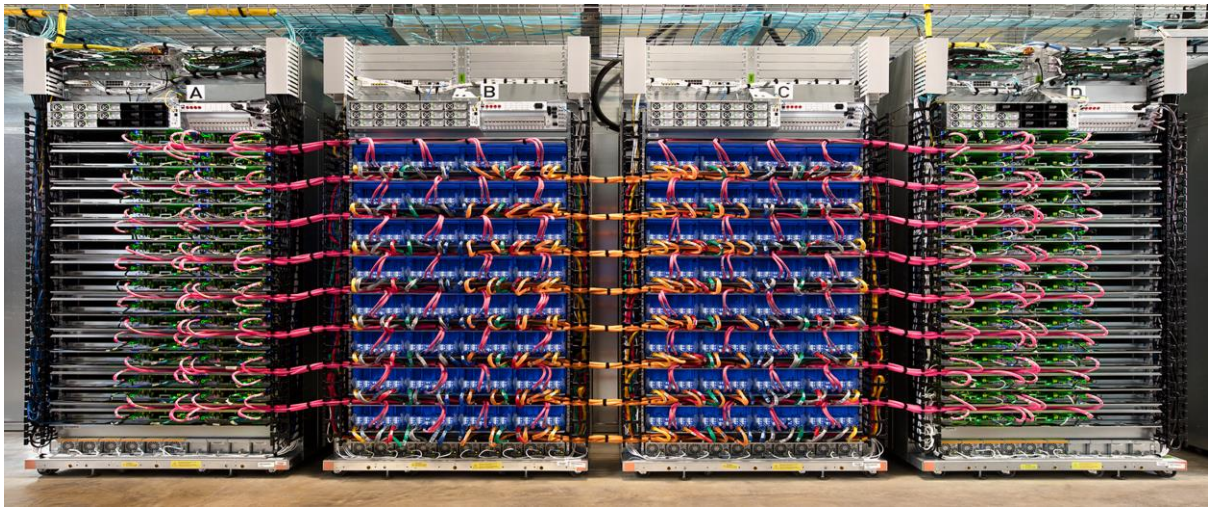

Figure 7: TPU Pods

4. Conclusion:

After going through the detail study about TPU we come to know that TPU's DNN applications use 8-bit integers rather than 32-bit floating point to improve efficiency of computation, memory bandwidth, and memory capacity.

The 2D organization enables systolic arrays, which reduce register accesses and energy

Because the TPU is a DSA, it can drop features required by CPUs and GPUs that DNNs don't use. Such omissions make the TPU cheaper, save energy, and allow transistors to be repurposed for domain-specific optimizations.

The TPU has one thread, while the K80 has 13 and the CPU has 18. A single thread makes it easier to stay within a fixed latency limit of our inference applications, as well as save energy.

5. References:

[1] N. P. Jouppi, C. Young, N. Patil, D. Patterson, Motivation for and Evaluation of the First Tensor Processing Unit, 2018

[2] N. P. Jouppi et al., "In-datacenter performance analysis of a tensor processing unit," In Proceedings of the 44th Annual International Symposium on Computer Architecture (ISCA), 2917, pp. 1–12, 2017.

[3] S. Markidis, S. Chien, E. Laure, NVIDIA Tensor Core Programmability, Performance & Precision, 2018.

[4] White Paper by E. B. Olsen, Proposal for a High Precision Tensor Processing Unit, 2017.

[5] https://cloud.google.com/tpu/docs/tpus

[6] https://keras.io/datasets/