Review Article- Google TPU

Submitted by:

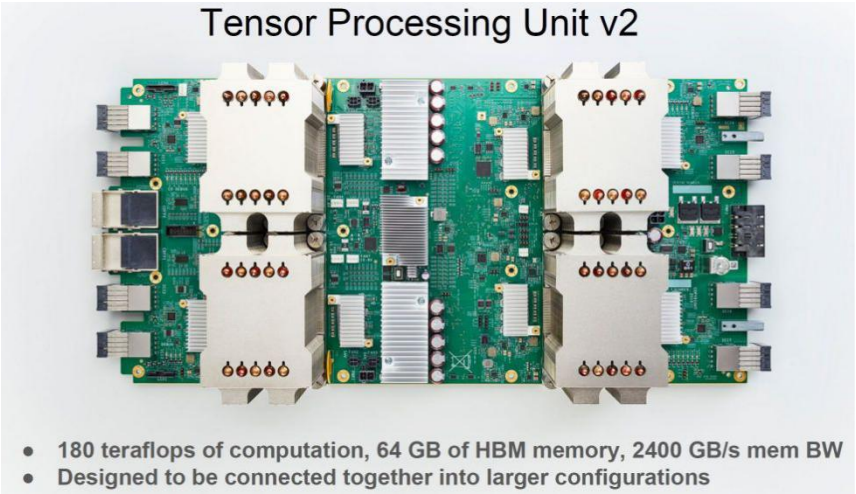
Ripunjay Narula(19BCE0470) and

Samvit Swaminathan(19BCE0629)

What is a TPU?

TPU stands for Tensor Processing Unit. It is an AI accelerator application-specific integrated circuit (ASIC). TPUs have been developed by Google in 2016 at Google I/O. However, TPUs have already been in Google data centers since 2015.

The chip is specifically designed for TensorFlow framework for neural network machine learning. Current TPU versions are already 3rd generation TPUs, launched in May 2018. Edge TPUs have also been launched in July 2018 for ML models for edge computing.

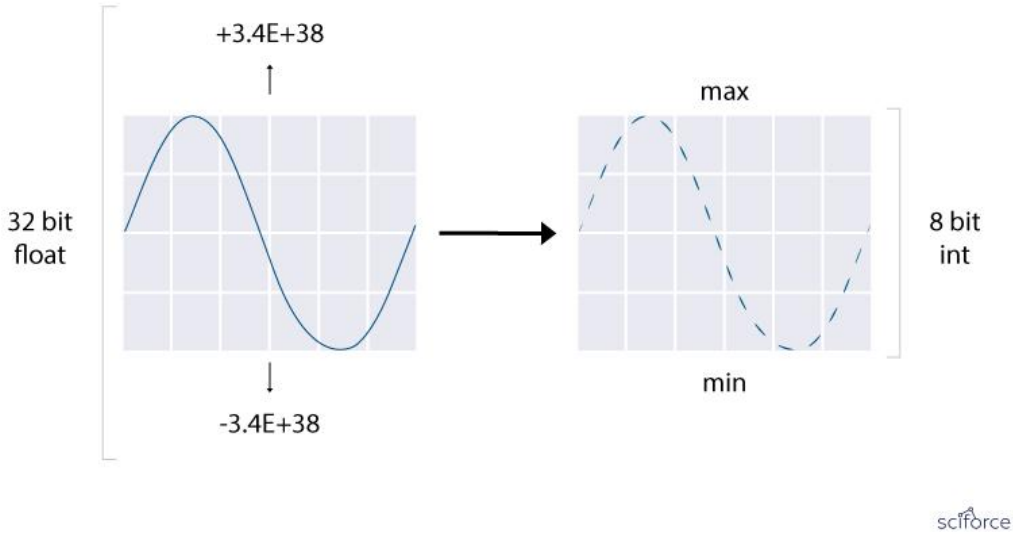


The TPUs have been designed, verified, built and deployed in just under 15 months, whereas typical ASIC development takes years.

How do TPU work?

1. Quantization

In line with the quantization technique, the process of approximation of an arbitrary value between a preset minimum and a maximum value with an 8-bit integer, TPUs contain 65,536 8-bit integer multipliers. In essence, this technique is compression of floating point calculations with 32-bit or even 16-bit numbers to 8-it integers. As you can see, the continuous large set of values (such as the real numbers) is converted to a discrete set (of integers) with maintaining the curve:



Quantization is the first powerful tool TPUs use to reduce the cost of neural network predictions without significant losses in accuracy.

2. Focus on inference maths

Secondly, the TPU design itself encapsulates the essence of neural network calculation. A 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.

They are controlled with a dozen high-level instructions that focus on the major mathematical operations required for neural network inference. A special compiler and software stack translate API calls from TensorFlow graphs into TPU instructions.

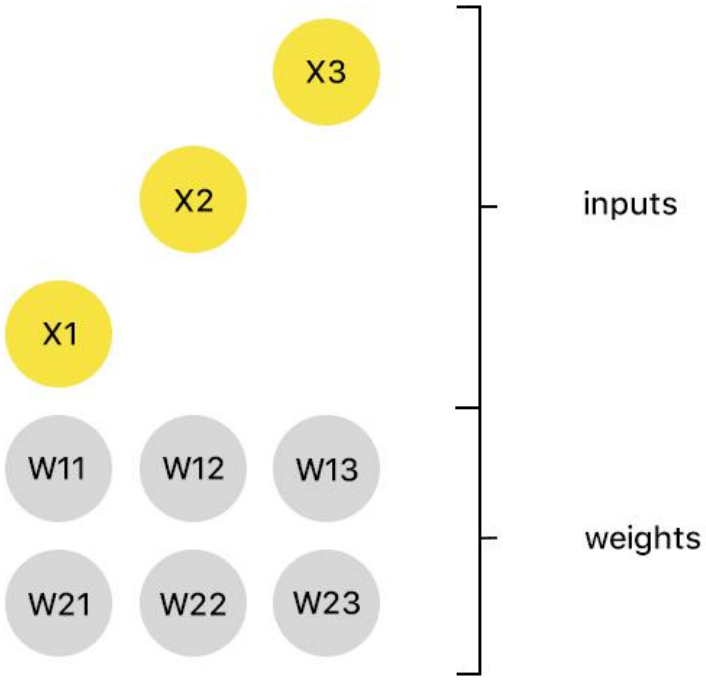
3. Parallel Processing

Typical RISC processors provide instructions for simple calculations such as multiplying by processing a single, or scalar, operation with each instruction. As you remember, a TPU contains a Matrix Multiplier Unit. It is designed as a **matrix, rather than scalar, processor** and processes hundreds of thousands of operations (= matrix operation) in a single clock cycle. Using such a matrix processor is like printing documents a whole page at a time rather than character-by-character or line-by-line.

4. A systolic array

The heart of the TPU is the new architecture of the MXU called a systolic array. In traditional architectures (such as CPUs or GPUs), values are stored in registers, and a program tells the Arithmetic Logic Units (ALUs) which registers to

read, the operation to perform (such as an addition, multiplication or logical AND) and the register into which to put the result. A program consists of a sequence of these operations. In an MXU, matrix multiplication reuses inputs many times to procduce the final output. A value is read once but used for many different operations without storing it back to a register. The ALUs perform only multiplications and additions in fixed patterns, and wires connect adjacent ALUs, which makes them short and energy-efficient.



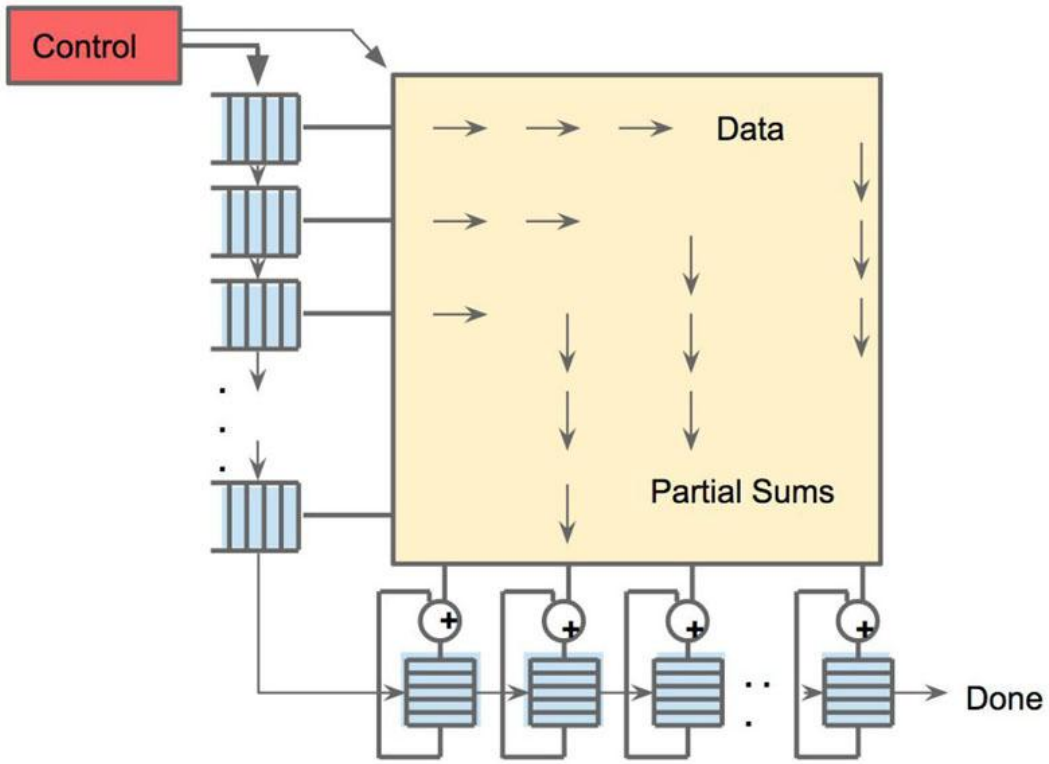
To understand this design, think of the heart pumping blood — like the data flowing through the chip in waves.

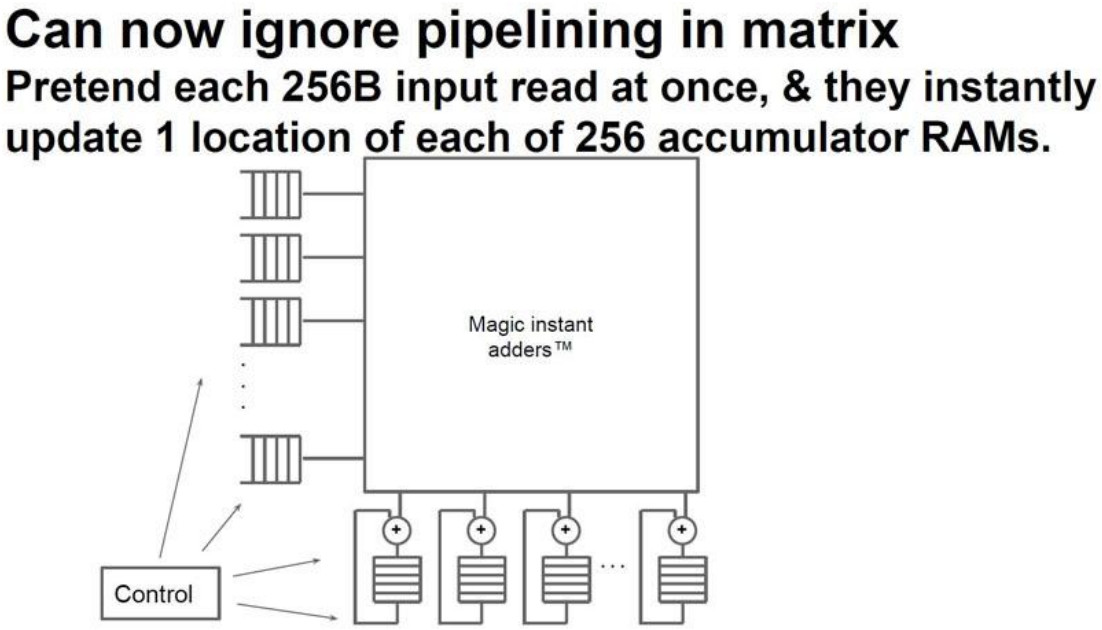
Google TPU Systolic Execution Diagram:

Memory bandwidth is extremely important in the architecture so the TPU is designed to efficiently move data around.

Matrix Multiplication Unit

- Contains 256 x 256 = 65,536 ALUs
- TPU runs at 700 MHz
- Able to compute 46 x 1012 multiply-and-add operations per second
- Equivalent to 92 Teraops per second in matrix unit





Google TPU Ignore Pipelining In Matrix

TPU instruction set

TPU is designed to be flexible enough to accelerate computation times of many kinds of neural networks model.

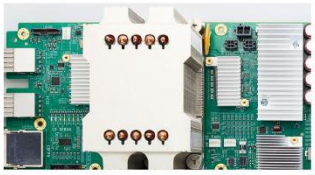
Modern CPUs are influenced by the Reduced Instruction Set Computer (RISC) design style. The idea of RISC is to define simple instructions (load, store, add, multiply) and execute them as fast as possible.

TPUs use Complex Instruction Set Computer (CISC) style as an instruction set. CISC focus on implementing high-level instructions that run complex tasks such as multiply-and-add many times with each instruction.

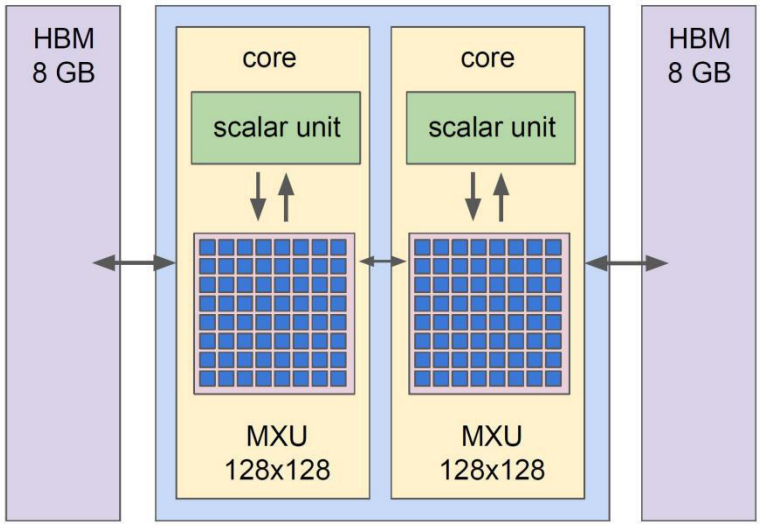
Cloud TPU v2 Architecture

How Google actually followed up the TPU is with the TPUv2. Instead of updating to higher bandwidth GDDR5, Google is using even faster HBM in 16GB capacity. That gives them 600GB/s of memory bandwidth. For comparison, that is about four sockets worth of AMD EPYC.

TPUv2 Chip



- 16 GB of HBM
- 600 GB/s mem BW
- Scalar unit: 32b float
- MXU: 32b float accumulation but reduced precision for multipliers
- 45 TFLOPS



What is a TPU made of?

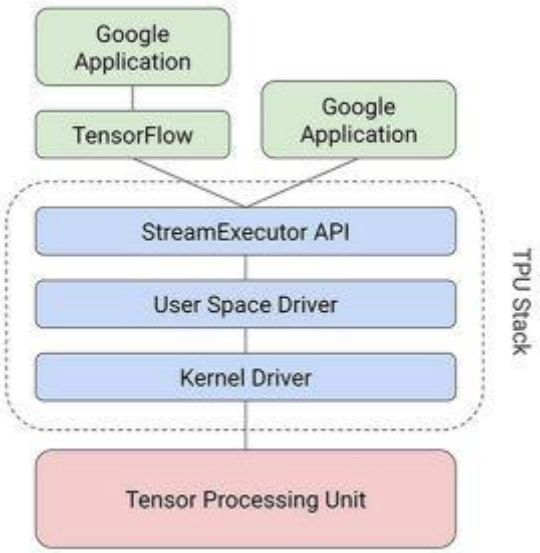
TPUs are made of several computational resources :

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

Here's an example of some high-levels instructions specifically designed for neural network inference that control how the MXU, UB, and AU work :

- Read_Host_Memory : Read data from memory
- Read_Weights : Read weights from memory
- MatrixMultiply/Convolve: Multiply or convolve with the data and weights, accumulate the results
- Activate : Apply activation functions
- Write_Host_Memory : Write results to memory

Google has created a compiler and software stack that translates API calls from TensorFlow graphs inti TPU instructions following this schema :

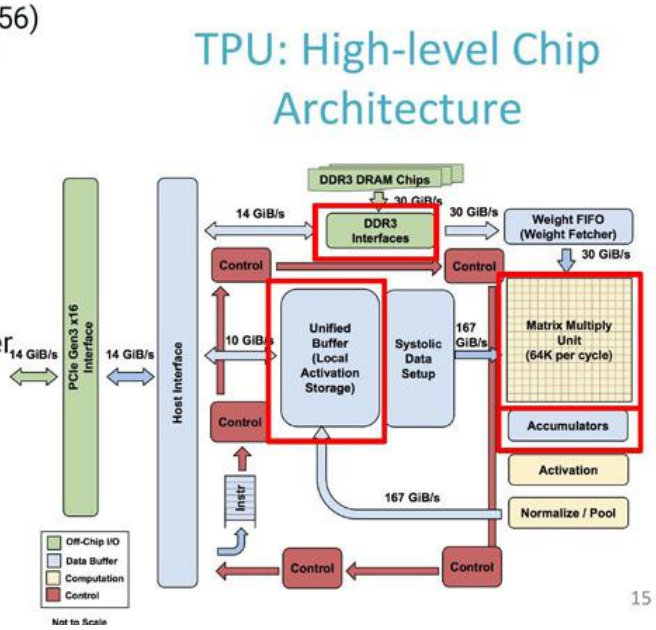


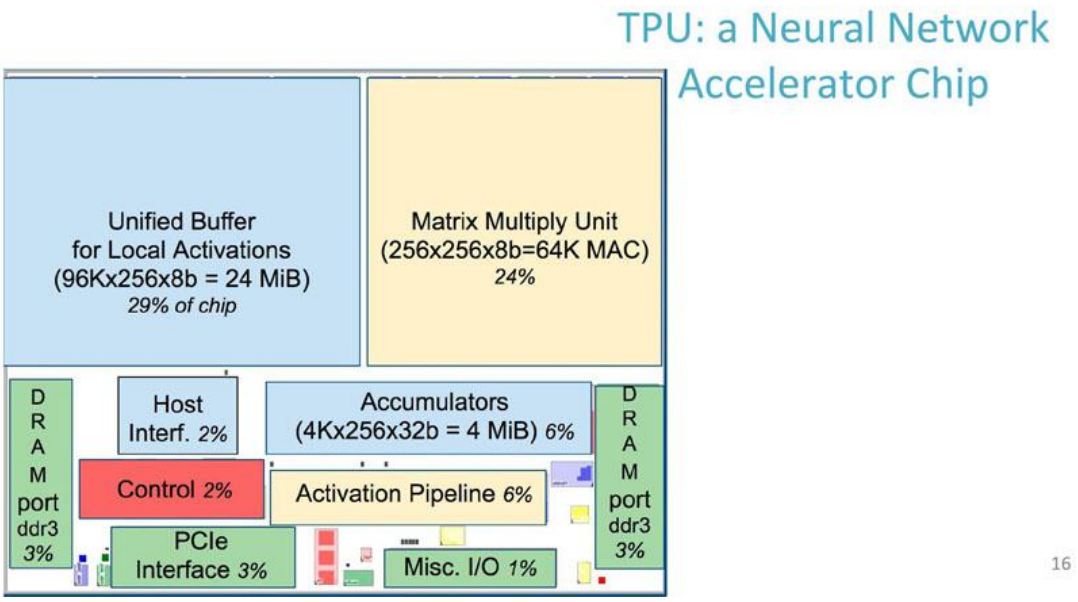
Specifically, it is designed for fast matrix multiply, important in deep learning applications.

Google TPU High-Level Architecture

Here is a view of how much logic is dedicated in the chip to different functions:

- The Matrix Unit: 65,536 (256x256) 8-bit multiply-accumulate units
- 700 MHz clock rate
- Peak: 92T operations/second
  - 65,536 * 2 * 700M
- >25X as many MACs vs GPU
- >100X as many MACs vs CPU
- 4 MiB of on-chip Accumulator memory
- 24 MiB of on-chip Unified Buffer (activation memory)
- 3.5X as much on-chip memory vs GPU
- Two 2133MHz DDR3 DRAM channels
- 8 GiB of off-chip weight DRAM memory



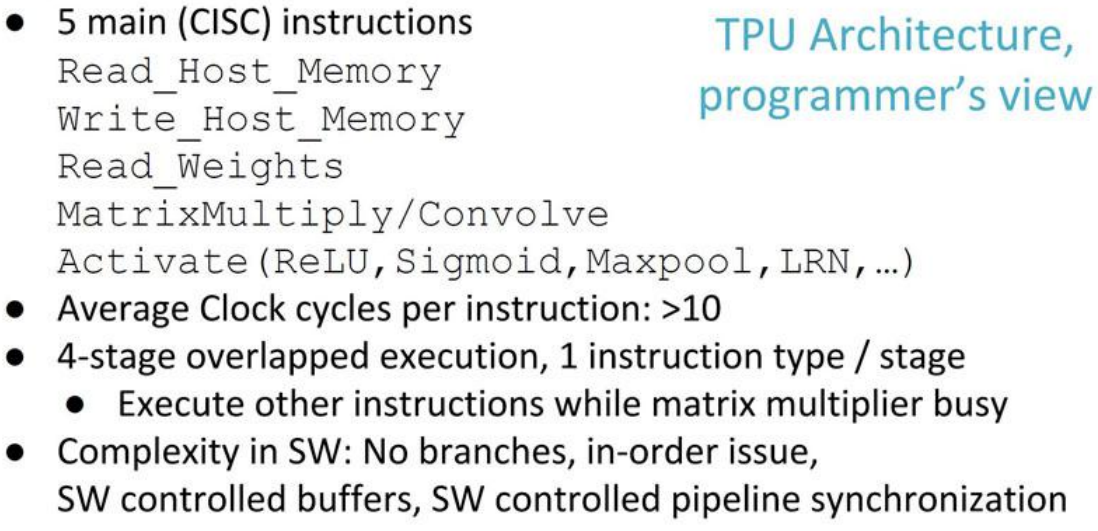


Google TPU Neural Network Accelerator Chip

To the programmer, Google is providing a view in terms of what using the TPU looks like.

Google TPU Architecture Programmer View

The TPU has only 11 instructions, five are commonly used. It is an in order design that relies on software to handle complexity.



What if there was a Google TPU (v1.1) Update GDDR5 for Bandwidth?

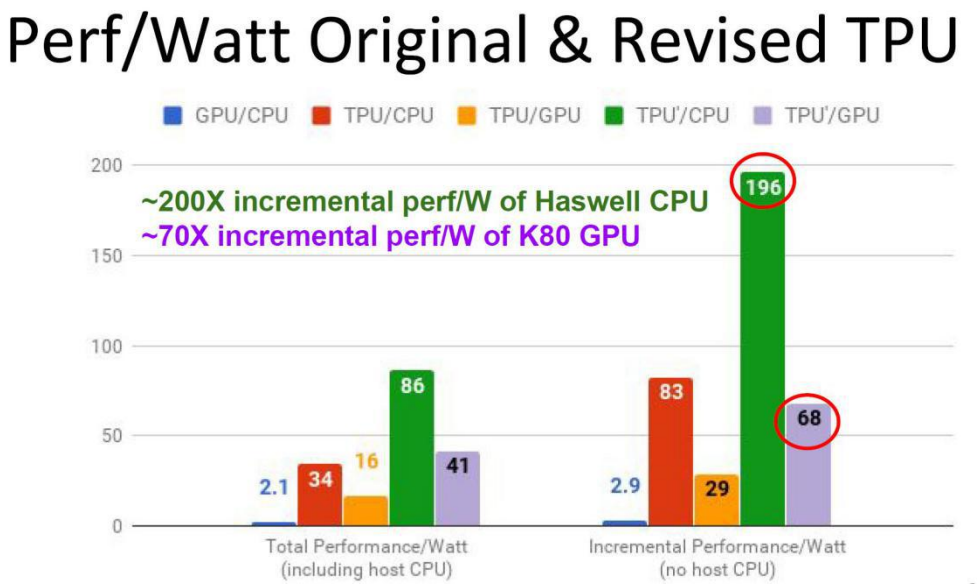
Google used DDR3 in its original design. One option was to move to DDR4 for higher bandwidth operation. Another potential design decision was to move to GDDR5 which is what Google showed at Hot Chips:

- Current DRAM
  - 2 DDR3 2133 ⇒ 34 GB/s
- Replace with GDDR5 like in K80 ⇒ 180 GB/s
  - Move Ridge Point from 1400 to 256

Improving TPU: Move "Ridge Point" to the left

Updated Google TPU GDDR5

The company explained that they were not necessarily compute limited in their original design. DDR3 was too slow. Here is what Google is claiming if the TPU had GDDR5 in terms of performance per Watt:



Updated Google Revised TPU GDDR5

The key here is that the GDDR5 version would have had more bandwidth and lower latency boosting performance. Instead of building the v1.1 version using GDDR5, Google leapfrogged GDDR5 and moved to HBM for its TPU v2 which could also be used for training, not just inference.

Cloud TPU v2 Architecture

How Google actually followed up the TPU is with the TPUv2? Instead of updating to higher bandwidth GDDR5, Google is using even faster HBM in 16GB capacity. That gives them 600GB/s of memory bandwidth. For comparison, that is about four sockets worth of AMD EPYC.

These are the much larger compute units with more TDP headroom that Google is making available to developers.

Google Cloud TPUs for ML Acceleration:

A Tensor is analogous to a numpy array and in fact uses Numpy. According to their documentation it is “NumPy is the fundamental package for scientific computing with Python. It contains among other things a powerful N-dimensional array object … ”
Arrays are the fundamental data structures used by machine learning algorithms. Multiplying and taking slices from arrays takes a lot of CPU clock cycles and memory. So Numpy was written to make writing code to do that easier. GPUs now make those operations run faster.
In particular, the math involved in doing ML includes adding and multiplying these objects:

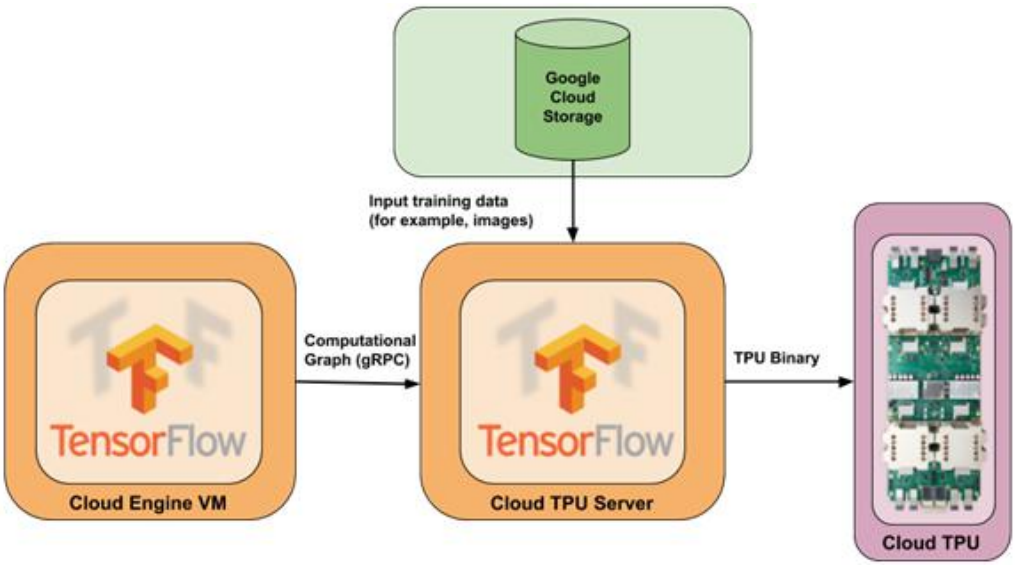
- scalar
- vector
- matrix

A CPU has 1 to 8 cores or more. A GPU has hundreds. The GPU and TPU are the same technology. The only difference is now selling it as a cloud service using proprietary GPU chips that they sell to no one else.
Google's approach to provisioning a TPU is different than Amazon's. At Amazon you pick a GPU-enabled template and spin up a virtual machine with that. Those templates all start with the letters **P3** and are listed here.

With Google you use their command line tool cptu to provide machines with TPUs. (And you can continue to use NVIDIA GPUs as well.)

According to Google's pricing information, each TPU cost $4.50 hour. Apparently they do not charge different rates for different TPU models even though they show three models on their website. That seems confusing as TPUs have different memory sizes and clock speeds. So one should be more expensive than another.

The TPU workload is distributed to what they call their **TPU Cloud Server**, as shown below.



An estimator is the tf.estimator.Estimator class. These are the implementation of neural networks, linear regression, and other objects with Python code that makes creating those kinds of objects simpler, since they leverage Numpy, Pandas, and other Python data structures and utilities.

Now Google says they have a TPU Estimator. You cannot download and use the GPU-enabled version of Tensorflow, which is different than regular TensorFlow in that it uses the CUDA SDK for that part of that code that is written in C and C++. There is no separate TPU-enabled version of TensorFlow. And unlike GPU, there appears to be no way to explicitly tell the code to use the TPU device, like in this code snippet that multiplies two matrices using GPU device **/device:GPU:n.** (To use the CPU you would write **/device:CPU:n,** where n can be any of the n CPUs on the computer.)

```
with tf.device('/device:GPU:0'):

    c = tf.matmul(x, y)
```

Scale Advantage?

One advantage of the TPU design would be that it lets you scale operations across different machines with their TPU servers. The user, of course, does not need to write any code to do that. This researcher has not yet studied how to do that with GPUs. In other words what do you do when your calculation runs out of memory? You can add PCI expansion cards '/device:GPU:1, 2, …., n or effectively do the same thing by paying Amazon for a larger template. But how do you implement something like a Mesos equivalent that would let you scale across a cluster of servers without having to hard-code device and server names? We will look at that and write you back.

USES

- RankBrain algorithm used by Google search
- Google Photos
- Google Translate
- Google Cloud Platform

Future Development

- Uses TPU version 2
- Each TPU include a high-speed network
- 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

Sources/Citations:

https://www.bmc.com/blogs/google-cloud-tpu/
https://medium.com/sciforce/understanding-tensor-processing-units-10ff41f50e78
http://meseec.ce.rit.edu/551-projects/fall2017/3-4.pdf
https://cloud.google.com/tpu/docs
https://www.servethehome.com/case-study-google-tpu-gddr5-hot-chips-29/

Survey Paper - Google TPU

Submitted by:
Ripunjay Narula(19BCE0470) and
Samvit Swaminathan(19BCE0629)

Understanding the Increasing Demands of Modern Deep Neural Networks

An understanding of Neural Network (NN) progression is important to understand the hardware being compared in this article. The current trend in NN programming is to create “deeper” NNs, called Deep Neural Networks (DNN). DNNs have increased the accuracy of NNs by reducing speech recognition errors, increasing the accuracy of image recognition, and beating human challengers in games such as Go and Jeopardy. This accuracy comes at a cost both when training a model and making predictions, called “inference.” These DNNs are trained on larger data sets and may have increasing number of input parameters.

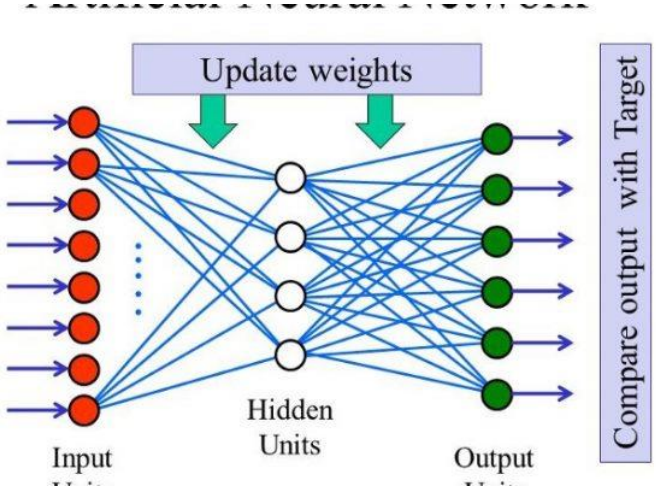
Inference is the term used when a trained NN is making predictions. Many companies use inference at scale so throughput, of course, is important, but also response time is essential. Think of how long your patience lasts when waiting for a response from Siri or Google Assistant when you asked for the nearest Italian restaurant. We will see the demand to reduce response time during inference is a powerful motive in modern NN hardware designs.

Each layer in a NN is a vector of floating point (FP) or integer numbers. Numbers used range from INT4, INT8, FP16, FP32 and FP 64. Larger FP provide greater accuracy, but they make the NN slow and more computationally expensive. Using 8-bits per neuron is becoming more common during certain stages as processing speed is largely reduced due to smaller bit sizes used in computation. Facebook is one example of having done research showing the benefits of using smaller bit size to optimize inference response-time.

The three factors:

1. Deeper Networks
2. Input Size
3. Big Data

are increasing the amount of compute required for machine learning. Considering this software problem, we can see an opportunity for hardware to step in and increase efficiency.

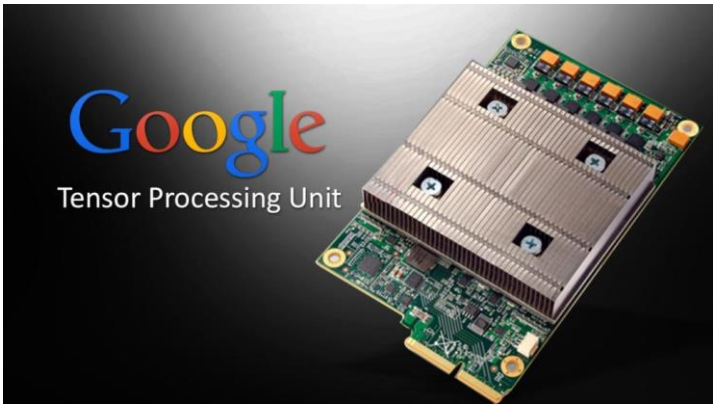


Hardware of Google Tensor Processing Unit (TPU)

for Machine Learning

Google's Tensor Processing Unit (TPU) is an application specific integrated circuit created by Google to process neural networks. Google announced, in 2016, the TPUs have been in use in their data centres for over a year. Google designed the hardware to work with their open-source software Tensorflow, an application specifically built for working with neural networks. One motivation for creating the Application-Specific Integrated Circuit (ASIC) was to support the growing number of speech translations Google continually processes.

The TPU works by creating a grid of simplified ALUs. The data is sent via a PCIe bus to the grid. As the multiplication and addition of each vector is applied to each layer the result is passed to the next layer creating a pipelining effect throughout the 128x128 matrix of ALUs. Memory requirements are low as the output of one layer of ALUs is the input of the next layer. This also reduces power consumption as memory access is more power expensive than ALU computation.



1. *Implementation and Cost*

The TPU is integrated into already running servers via a PCIe bus. This makes implementing it for Google inexpensive and quickly available

for rental on the cloud for between $1.35/hr and $5.22/hr! Multiple TPU pods can work together to speed up training models.

The TPU is a simple solution that does not handle “caches, branch prediction, out-of-order execution, multiprocessing, speculative prefetching, address coalescing, multithreading, context switching, and so forth. Minimalism is a virtue of domain-specific processors.” Google wanted a pluggable solution that could be implemented quickly and with little risk/cost.

2. *Tensorflow Software*

Google’s hardware is optimized for Tensorflow. This is an open-sourced library built for building neural networks and was released in 2015.

3. *TPU hardware specifications*

The TPU instructions are sent via PCIe Gen3 x16 bus into an instruction buffer. The TPU does *not* fetch instructions from the CPU, rather the CPU will send instructions to a buffer. This was designed to “keep the matrix unit busy”.

It also reduces the number of pipeline steps required. The architecture uses a 4-stage pipeline, though the grid of ALUs have their own pipelining during matrix computation. The pipelining available thus varies greatly due to various NN instructions that may require 1000s of clock cycles. Twenty-three percent of stalls occur due to read-after-write (RAW) dependencies.

Even though memory is kept low as ALU passes data to contiguous ALU vectors, memory bandwidth is an optimization problem that many NN (LSTM and MLP) bottleneck on. Google’s NN analytics show peak performance of certain models is not reached because of memory constraints. While detrimental to peak performance adding

more memory is a known issue that can be solved by simply adding more memory (cheap) or some larger L2/L3 caches.

The MMU contains 256x256 MACs that perform 8-bit to 64-bit multiplication and addition on signed and unsigned integers. It can increase accuracy of computation by increasing to 16-bit or 32-bit calculations at the cost of speed and number of items in the vectors being processed. It can write 256 8-bit values per clock cycle.

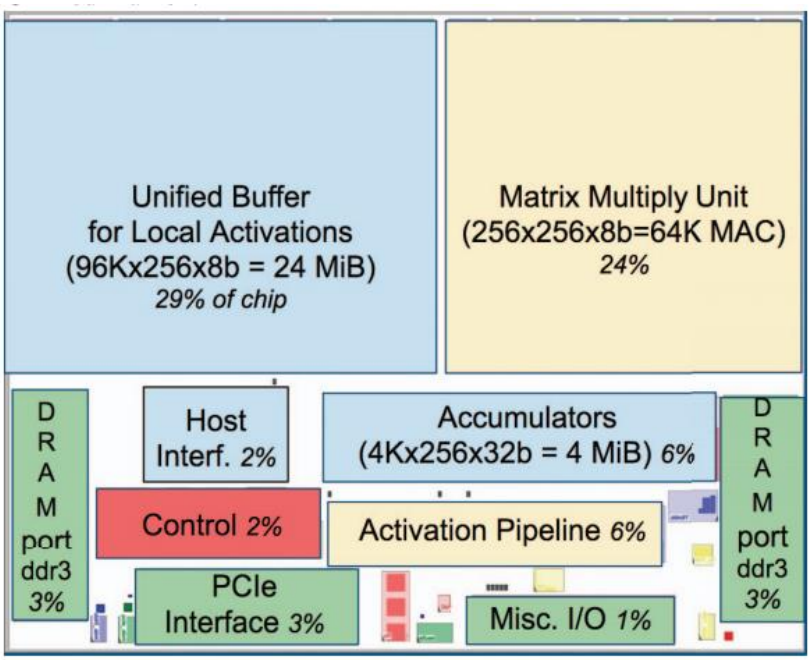


Figure 2. Floorplan of TPU die. The shading follows Figure 1. The light (blue) datapath is 67%, the medium (green) I/O is 10%, and the dark (red) control is just 2% of the die. Control is much larger (and much harder to design) in a CPU or GPU.

4. TFLOPS Performance

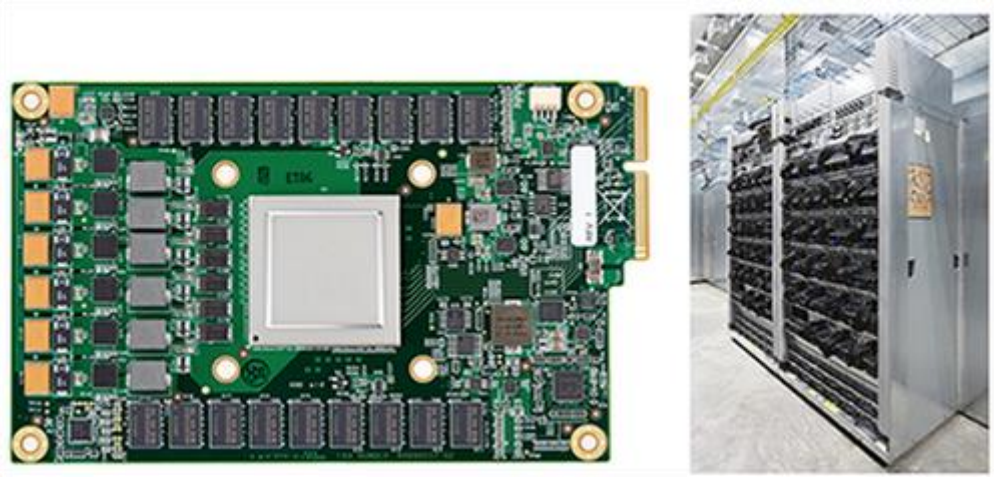
Google claims the TPU is 15–30 times faster at inference than the NVIDIA K80 GPU. The TPU has 25 times as many multiplier accumulators (MACs) and 3.5 times as much on-chip memory as the K80 GPU. They also claim the energy performance is 30–80x more efficient than “contemporary CPUs and GPUs. Google’s TPU reaches 180 TFLOPS when combined via four 45 TFLOPS chips.

5. *Power*

Power optimizations are achieved in part by reducing the reads and writes on the buffer and memory. The TPU idle power required is high compared to GPU and CPU. At 10% capacity the wattage used is 88% the power it uses at 100% capacity.

6. *TPU Pods*

Each board can be connected together to form "'multi-petaflop' ML supercomputers that [they] call 'TPU Pods''. Google claims machine learning (ML) models can be trained overnight, rather than waiting days or weeks to train a business-critical model. Google claims training ResNet-50[A] and Transformer ML models drop from the about a day to under 30 minutes on a full TPU pod.

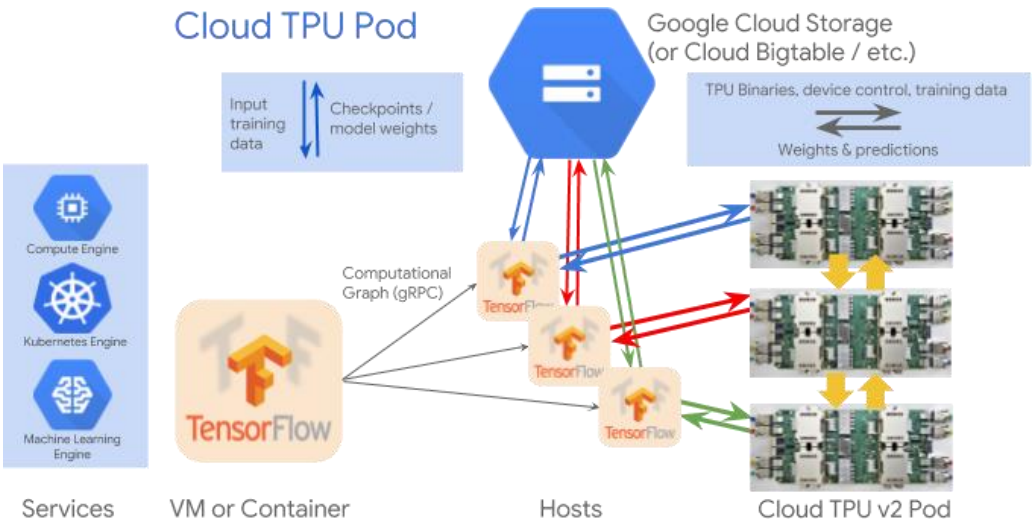


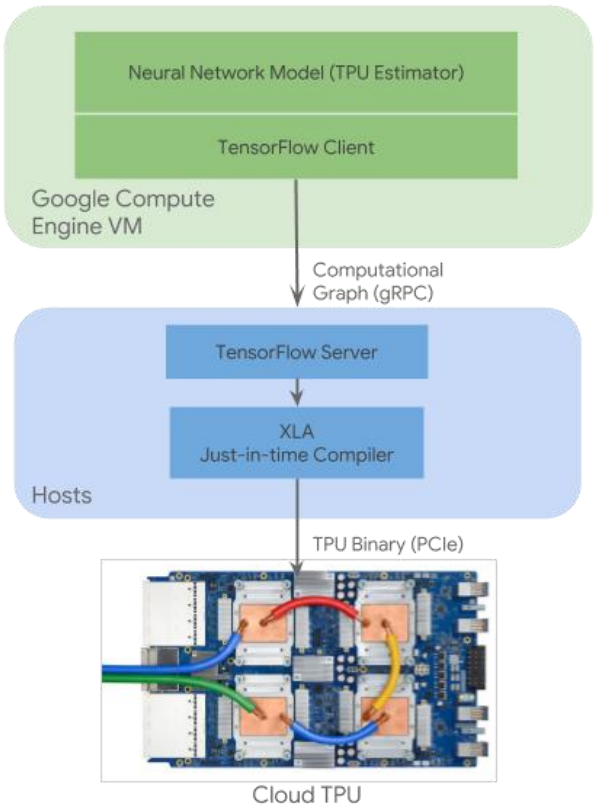
Google's first Tensor Processing Unit (TPU) on a printed circuit board (left);

TPUs deployed in a Google datacenter (right)

By evaluating a custom ASIC---called a Tensor Processing Unit (TPU)---deployed in datacentres since 2015 that accelerates the inference

phase of neural networks (NN). The heart of the TPU is a 65,536 8-bit MAC matrix multiply unit that offers 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 requirement of our NN applications than are the time-varying optimizations of CPUs and GPUs (caches, out-of-order execution, multithreading, multiprocessing, prefetching etc.) that help average throughput more than guaranteed latency. The lack of such features helps explain why, despite having myriad MACs and a big memory, the TPU is relatively small and low power. Their workload was written in the high-level TensorFlow framework, uses production NN applications (MLPs, CNNs, and LSTMs) that represent 95% of our datacenters' NN inference demand. Despite low utilization for some applications, the TPU is on average about 15X - 30X faster than its contemporary GPU or CPU, with TOPS/Watt about 30X - 80X higher. Moreover, using the GPU's GDDR5 memory in the TPU would triple achieved TOPS and raise TOPS/Watt to nearly 70X the GPU and 200X the CPU.





Google is offering more powerful hardware for its machine learning cloud customers as it launches its next generation of Tensor Processing Unit (TPU) chips through its Google Compute Engine.

Officially called Cloud TPUs, these chips can work with Intel's Skylake processors, as well as Nvidia's GPUs. The new generation of TPUs offer 180 teraflops of floating-point performance, and Google has designed these chips so that they can be stacked in what the company calls a TPU pod, which houses 64 TPUs and can provide up to 11.5 petaflops of compute power.

Taken together, these TPUs are designed to accelerate machine learning, while giving customers access to the power of the Google's cloud, which can lower the barrier to entry for many businesses trying to build machine learning and artificial intelligence (AI) applications.

- **A TPU pod**

AlphaGo is powered by TPU, which is built on a 28nm process, runs at 700MHz and consumes 40W when running. Because TPU is needed to deploy to Google's existing servers as fast as possible, the company chose to package the processor as an external accelerator card that fits into a SATA hard disk slot for drop-in installation. The TPU is connected to its host via a PCIe Gen3 x16 bus that provides 12.5GB/s of effective bandwidth. The secret of TPU's outstanding performance is its dedication to neural network inference. The quantization choices, CISC instruction set, matrix processor and minimal design all became possible when it was decided to focus on neural network inference. Google also announced that its second-generation TPUs were coming to Google Cloud to accelerate a wide range of machine learning workloads, including both training and inference.

Cloud TPU is designed to run cutting-edge machine learning models with AI services on Google Cloud. And its custom high-speed network offers over **100 petaflops** of performance in a single pod — enough

computational power to transform your business or create the next research breakthrough.

What Did Google Announce?

Google announced a new ASIC that will accelerate its internal machine learning algorithms, as well as provide a compelling platform for AI practitioners to use the Google Cloud for their research, development, and production AI work. The 2nd generation TPOU chip delivers 45 Trillion Floating Point Operations Per Second (presumably 16-bit TFLOPS) for Machine Learning, roughly twice that which is available today from NVIDIA P100 (20 TFLOPS) or Advanced Micro Devices' upcoming Vega GPU (25 TFLOPS), however it will be surpasses by NVIDIA's new Volta chip described below. The "Cloud TPU" is packaged on a 4-chip module complete with a fabric to interconnect these powerful processors, allowing very high levels of scaling. This scaling capability is important, because training a neural network can take advantage of an almost limitless supply of accelerators.

The 4 chip Cloud TPU board forms the building block node for interconnecting 1000s of TPUs in a cluster for research and cloud services. There were no visible signs of active cooling, and the company did not disclose power consumption details. (Source: Google). The 4-chip Cloud TPU, therefore delivers 180 TFLOPS, and were shown in a "TPU Pod" with 32 interconnected boards, delivering 11.5 TeraFlops of peak performance—effectively a large supercomputer in a single rack.

Google also announced the TensorFlow Research Cloud, a 1,000-TPU (4,000 Cloud TPU Chip) supercomputer delivering 180 PetaFlops (one thousand trillion, or one quadrillion, presumably 16-bit FLOPS) of

compute power, available free to qualified research teams. While this is similar but significantly larger in concept to the Saturn V Supercomputer from NVIDIA, the Google Supercomputer is designed to support only Google's own open-source TensorFlow Machine Learning framework and ecosystem, while Saturn V is available for all types of software.

Conclusions

Google is attempting to build a dominant position in Artificial Intelligence, from optimized search services, to Android capabilities, to autonomous vehicles. Having complete control of the required technology stack enables it to optimize its technology while lowering its CapEx compared to buying technology from the outside. The synergies of controlling the Cloud TPU and TensorFlow should give the company a strategic competitive advantage, while accelerating the underlying science of AI and the industry at large. The impact on NVIDIA may be real, but relatively contained, at least for now. The longer-term impact on the GPU for AI, which has built NVIDIA's impressive growth engine, remains to be seen.