



Systematic Analysis on TPU's History and Key Technologies Behind It

Yilin Zhao

Ashford School, Ashford, TN24 8PB, UK
ZhaoJ@ashpupil.co.uk

Abstract. With the rapid growth of artificial intelligence and deep learning, traditional computing architectures face limitations in efficiency and scalability. This paper explores Google's Tensor Processing Unit (TPU) to understand how specialized hardware accelerates AI workloads and drives next-generation computational innovations. This article analysis the development and key technologies behind Google's TPU. It begins by explaining how Google created the TPU to meet rising computational demands for deep learning, which CPUs and GPUs could not efficiently handle. The paper outlines the main components of a TPU, such as the systolic array, high-bandwidth memory, and matrix multiply unit. It then traces the evolution of TPU versions from v1 to v4, highlighting improvements in speed, cooling, and scalability. Key technologies like matrix multiplication optimization, low-precision computing, and the use of TensorFlow are also discussed. Finally, the impact of TPUs on both Google's services and broader AI research is summarized, showing how they make high-performance computing more accessible and efficient.

Keywords: TPU, Systolic Array, Machine Learning, Deep Learning

1 Introduction

The idea of TPU was first introduced in 2013, the scene was Google need to double the datacentre to meet the demand of computation for 3 minutes a day using speech recognition DNNs. However, it would be very expensive to satisfy with the conventional CPUs. Therefore, a custom ASIC chip became what Google looking for [1]. In the same year, TPU project started by Norman Jouppi and his team.

The foundation of Deep Learning---Neural Network was built in late 90s: The Convolutional Neural Networks (CNNs) in 1995; Long Short-Term Memory (LSTM) in 1997; Deep Belief Networks (DBNs) in 2006. In the following decade after these key concepts were released, the deep learning breakthrough. Including Alex Net in 2012; Generative Adversarial Networks (GANs) in 2014; Res Net in 2015.

The major demand of deep learning's training of Google is to provide enough computation for datacentre. GPU fit in small scale and irregular computation so that datacentre-scale is clearly out of range. CPU fit in extreme batches FC model. For example, the recommendation system of Facebook. But software design of CPU is

highly mature so less dramatic gain in the future. The advantage of TPU is that in training CNN and RNN, TPU have 2.2 and 3 times more usage of floating-point operations per second (FLOPS). TPU is also fit in batches training and batches CNN. Those characteristic shows irreplaceability of TPU in training AI [2].

2 Typical architectures

A standard TPU chip contain 6 key architecture and hardware: systolic array; High-Bandwidth Memory; Unified Buffer; Activation Unit; Matrix Multiply Unit; Compiler.

2.1 Systolic array

Systolic array is a specialized data processing unit. There are some key defining characteristics including synchrony that all processing elements are synchronized by a global clock and each PE perform a fixed operation; the data flows between PEs and pipeline in rhythmic pattern; temporal locality make sure data is reused as it passes through multiple PEs so that time is saved by data reusing; spatial locality means every PE only connected with its neighbours so smaller scale communication network compare with tradition system; all PEs work concurrently and new data is ingested every cycle and results flow out every cycle.

2.2 High-Bandwidth Memory (HBM)

HBM is a high-performance RAM unit, not only TPU, but also used in GPUs. It uses 3D stacking, extremely wide I/O Bus and 2.5D integration technics to achieve such high bandwidth. Another advantage of HBM is the location difference between HBM and traditional memory: the HBM is integrated in the processor, which means it solve the Memory wall problem. Traditional processor can compute data far faster than that can fetch from the standard memory. The time of waiting for data wasted their immense computational power [1, 3].

2.3 Unified Buffer (UB)

UB is a cache but not only a cache, it is the primary on-chip memory for housing the entire matrix of intermediate computational results. It is the key role of data reusing. After a loop the data is not sent back to the host DRAM but UB. If the workload is suitable, the time saved is a 100x to 1000x speedup [1].

2.4 Matrix Multiply Unit (MXU)

MXU is a specialized hardware block take place of traditional ALU in TPU. It only used to calculate massively parallel matrix multiplication. Basic structure of MUX is shown in Fig.1.

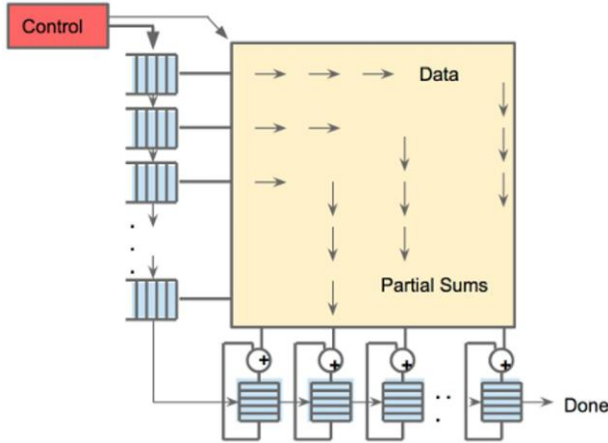


Fig. 1. Basic structure of MUX [1]

The massive parallelism provides best fit hardware condition for systolic array and the characteristic of fixed- function don't need to fetch, decode and schedule instructions for each operation. The hardware's physical layout defines the program for matrix multiplication [1].

2.5 Compiler

Compiler is an essential translator used to breakdown machine learning model into a TPU friendly program. Without compiler, TPU cannot do any computation.

3 Evolution of TPU

3.1 TPU v1

TPU v1 was designed to do inference only, it was deployed in 2015 and did AlphaGo's calculations. It used a systolic array of 256x256 ALUs. It can perform 65,536 8-bit multiply-and-add operations every cycle; A large 8 GiB DRAM are used to store the pre-trained model weights.

The UB it used to be a 24 MiB SRAM to be a register file to store intermediate inputs and outputs. The performance of TPU v1 was 15x to 30x faster than GPU and CPU; the power-efficient is 30x to 80x compared to GPU and CPU. However, the limitations of TPU v1 were obvious: TPU v1 was only designed to run trained models. It lacks high-precision arithmetic and hardware features to do model training [4].

3.2 TPU v2

TPU v2 was deployed in 2017 and publicly announced in 2018. Its evolute forms single inference chip into a multi-chip module that can operate both inference and training. TPU v2 introduced a dedicated high-speed interconnect fabric to link multiple chips together. What's new is a vector processing unit was added, it can process non-matrix

operations that are common in neural network training so that the MXU was free to focus on its specialty.

The idea of TPU pod was being made, all chips can communicate directly with each other, pod is able to split a large batch or model across chips. Each TPU v2 core delivers 45 TFLOPS in BF16 and FP16 mixed precision. Full pod performance (256-core) has a theoretical peak performance of 11.5 petaflops. TPU v2 was used to train Google's BERT. The time required reduced from weeks to hours. The pod design also allow outer costumer can use Google cloud to train models [5].

3.3 TPU v3

TPU v3 was announced in 2018 and deployed in 2019. It focuses on increasing performance rather than update architectural design. Each chip has 32GiB of HBM. The most important update is the liquid cooling system, this design means TPU pod can operate at a much higher thermal design power.

The system TPU v3 used was a form of direct-to-chip liquid cooling. Each TPU v3 board has a cold plate that sits directly on the ASICs. The cooling liquid used is the mixture of glycol and deionized water. The reasons of using glycol are mostly lowering the freezing point and raising the boiling point of the liquid. Glycol prevents the water freeze, expand and rupture the pipes. On the other hand, glycol can provide a higher margin of safety against the coolant boiling if local hot spots on the chip cause extremely efficient heat transfer, though this is a lesser concern in a well-designed system [6].

3.4 TPU v4

TPU v4 was announced in 2021. A key highlight of this generation is v4's pod would be about 90% carbon-free energy matched. A full TPU v4 pod has 4096 chips which is four times larger than a full TPU v3 pod. Compared to TPU v4, v3 used a fixed 2D torus network. Optical circuit switch (OCS) is a remarkable innovation which make TPU v4 more powerful than TPU v3. A single TPU v4 chip offers more than 2x the peak performance of a TPU v3 chip. The OCS can dynamically reconfigure the connections between thousands of TPU chips in microseconds.

The biggest advantage is the flexibility, resiliently, and efficiency. If a different network topology is more efficient for any ML model, the system can reroute traffic around to without losing a step. Another technic involved is sparse core. It's common in large-scale recommendation system, for example Google search. Based on this, a new concept of "Co-Design with AI Models" was raised as a fundamental philosophy for TPU project. This idea effects many models design such as T5, GPT-3 and BERT [7].

4 Key technologies

4.1 Matrix Multiplication Optimization (MMO)

MMO is the process of transforming the naive algorithm to drastically improve its performance. The primary goal is to maintain the number of floating-point operations and improve computation's memory efficient. Key techniques involved are loop ordering; loop blocking; parallelization and Bfloat16 precision. The naive order may stride through memory non-contiguously, causing poor cache utilization. Reordering the loops can ensure inner loops access contiguous blocks of memory, which is cache friendly.

Instead of processing entire rows and columns, the matrices are divided into small blocks so once a block is loaded into fast cache, all operations on theta block are performed before moving to the next. This led to the maximum size of data reusing. Bfloat16 has the same 8-bit exponent as a standard float32, this makes it easy for TPU to convert float32 models into float16 without extensive rescaling. The smaller 16-bit size means twice the data can be moved from memory and stored in registers compared to float32, effectively doubling the bandwidth and cache efficiency for the same silicon cost. The reduced precision is often sufficient for the gradient calculations in neural network training [8].

4.2 Low-Precision Computing

Low-precision computing is a foundational pillar of the TPU and its exceptional performance for machine learning workloads. It refers to the use of numerical data formats that use less bits than traditional FP32 to represent numbers. The core thesis is that neural network inference and training are numerically robust and can tolerate the reduced precision of lower-bit arithmetic.

This tolerance allows hardware designers to trade numerical precision for massive gains in three critical areas: computational throughput; memory bandwidth and power consumption. Smaller datatype can increase the number of operations per cycle. As the bit-width halved, the effective bandwidth immediately doubled. On the other hand, moving fewer bits across the chip and performing arithmetic on smaller circuits consumers significantly less energy [8].

4.3 Tensor flow

TensorFlow is an open-source software library designed for high-performance numerical computation, forming the foundation for developing and deploying machine learning models. Its core architecture utilises a dataflow graph paradigm, where nodes represent mathematical operations and edges represent the multidimensional data arrays, or tensors, that flow between them.

This structure allows for efficient execution and optimisation across various hardware platforms, including CPUs, GPUs, and TPUs. A key feature is it define-then-run execution model, where users first construct a computational graph defining all operations and their dependencies. This graph is subsequently executed within a session, which handles resource allocation and translation into optimised low-level

kernels. The framework also supports eager execution for immediate evaluation, enhancing debugging and prototyping.

For production, the function decorator converts Python functions into static graphs, balancing usability with performance. Training is facilitated through automatic differentiation via the gradient tape mechanism, which calculates gradients for optimisers to adjust model parameters. The high-level Keras API simplifies model building by providing modular layers and pre-implemented components. Supported by an extensive ecosystem for data handling and deployment tools, TensorFlow offers a comprehensive environment for the entire machine learning workflow.

5 Applications and Impact

5.1 Google's internal use

The computational demands of processing billions of search queries and translating vast amounts of text in real-time are met by the TPU's architecture, which is optimized for the low-precision, high-throughput matrix operations fundamental to deep learning. TPU can analyse thousands of features. For instant, query context, user history and webpage content.

On the other hand, TPU boosted Google's neural machine translation system (GNMT) as well. Which include Google Translate and Model Training Service. Training large sequence-to-sequence models with attention mechanisms is an immensely computationally intensive task. TPUs drastically reduce the training time for these models, allowing for more rapid experimentation with larger architectures and bigger datasets. Furthermore, during inference, TPUs enable the low-latency execution of these complex models, facilitating real-time translation across numerous language pairs.

5.2 Impact on research and industry

As TPU v2's release, TPU pods had introduced to the market, this has levelled the playing field. A small AI startup or a university research lab can now rent a slice of a supercomputer for a fraction of the cost of building one, enabling them to tackle problems previously thought impossible without massive capital [9]. The cost effective in training and deploying large models. It makes iterative research faster and cheaper and enables the deployment of more accurate models in production environments where cost was a limiting factor [10].

Google developed and popularized the bfloat16 (Brain floating point) number format to maximize TPU efficiency. It provides the dynamic range of a 32-bit float with the storage size of a 16-bit float. Nowadays, bfloat16 is an industry standard. Even Intel, Arm and AMD have all incorporated bfloat16 to support into their upcoming CPU architectures. This widespread adoption improves AI performance across the entire hardware ecosystem, not just on TPUs [8]. The idea of hardware-software co-design is advanced. This philosophy has been adopted by the entire industry. It proved that the future of high-performance AI computing lies not in general-purpose chips but in

specialized accelerators (ASICs) tuned for specific workloads. This has spurred a new golden age of computer architecture, with countless companies now developing their own AI chips.

6 Conclusion

This article lists out advantages of TPU and the current situation. This article can be guidance and reference in the problem of whether companies should or should not rent TPU for business. The development of the TPU represents a significant milestone in the evolution of specialized hardware for artificial intelligence. Key architectural innovations such as the systolic array, high-bandwidth memory, and dedicated matrix multiplication units have enabled TPUs to achieve remarkable speedups and power savings compared to general-purpose CPUs and GPUs. Moreover, the adoption of low-precision computing and advanced compilation techniques has further optimized computational throughput and memory efficiency, making TPUs exceptionally suited for both training and inference of large-scale neural networks. The introduction of TPU pods and optical circuit switching in later generations democratized access to supercomputing-scale resources, enabling broader academic and industrial research. As AI models grow and complexity, the role of specialized accelerators like TPUs will become increasingly critical, continuing to drive innovation in high-performance computing and artificial intelligence.

References

1. Jouppi, N.P., Young, C., Patil, N., et al.: In-datacenter performance analysis of a tensor processing unit. In: Proceedings of the 44th Annual International Symposium on Computer Architecture, pp. 1–12 (2017)
2. Wang, Y.E., Wei, G.Y., Brooks, D.: Benchmarking TPU, GPU, and CPU platforms for deep learning. arXiv preprint arXiv:1907.10701 (2019)
3. Zhu, M., Zhuo, Y., Wang, C., et al.: Performance evaluation and optimization of HBM-enabled GPU for data-intensive applications. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems* 26(5), 831–840 (2018)
4. Jouppi, N., Kurian, G., Li, S., et al.: TPU v4: An optically reconfigurable supercomputer for machine learning with hardware support for embeddings. In: Proceedings of the 50th Annual International Symposium on Computer Architecture, pp. 1–14 (2023)
5. Carrión, D.S., Prohaska, V.: Exploration of TPUs for AI applications. arXiv preprint arXiv:2309.08918 (2023)
6. Salim, M.: Quantum revolution: Redefining industry and the global chip race. *Quantum* (2025)
7. Jouppi, N., Kurian, G., Li, S., et al.: TPU v4: An optically reconfigurable supercomputer for machine learning with hardware support for embeddings. In: Proceedings of the 50th Annual International Symposium on Computer Architecture, pp. 1–14 (2023)
8. Wang, S., Kanwar, P.: Bfloat16: The secret to high performance on cloud TPUs. <https://cloud.google.com/blog/products/ai-machine-learning/bfloat16-the-secret-to-high-performance-on-cloud-tpus>, last accessed 2021

9. Chowdhery, A., Narang, S., Devlin, J., et al.: PaLM: Scaling language modeling with pathways. *Journal of Machine Learning Research* 24(240), 1–113 (2023)
10. Armoni, M.: Tensor processing units (TPU): A technical analysis and their impact on artificial intelligence. *Tech4Future Information Technology Report* (2023)

Open Access This chapter is licensed under the terms of the Creative Commons Attribution-NonCommercial 4.0 International License (<http://creativecommons.org/licenses/by-nc/4.0/>), which permits any noncommercial use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.

