EE 451: Supervised Research Exposition Industry Mentored Project

Guide: Mr. Sunil Shenoy
Co-Guide: Prof. Laxmeesha
Presenter: Rishabh
Roll Number: 200260041

Motivation

- The main goal of this project is to study the architecture of commercial accelerators
- Additionally I intend to perform a comparative study between accelerators developed for similar applications

Overview of Accelerator Surveys

- Focus on gathering a comprehensive list of AI accelerators
- This includes their computational capability, power efficiency, and the computational effectiveness in embedded and data center applications.

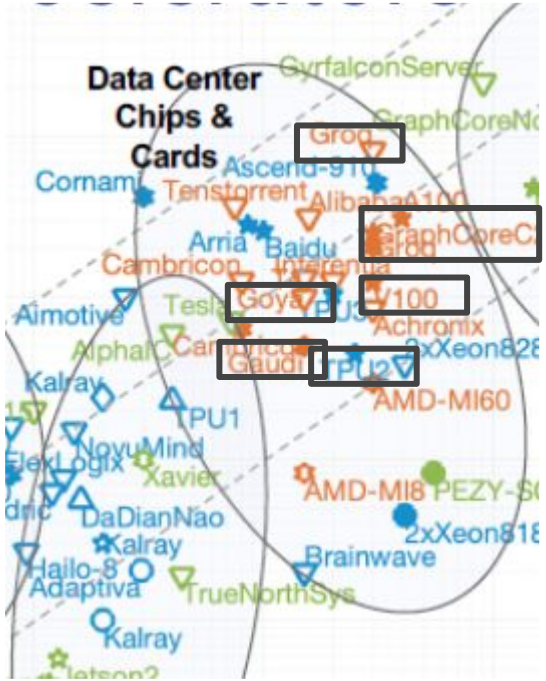
Classification Based on Technology

- Another factor used to classify accelerators was the technology used
- They are PIM, Many core, dataflow, CPU, FPGA, and multicore

The work ahead…

Literature Survey

1. Nvidia Hopper
2. Nvidia Ampere
3. Nvidia Turing
4. Habana Labs Gaudi
5. Habana Labs Goya
6. Groq TSP
7. Google TPU



How do we compare?

- Comparison done based performance and power consumed.
- Performance is measured in terms of GOps/s
- Accelerators are designed specific to their applications, and it would be more appropriate to group them according to their purpose
  - Very Low Power
  - Embedded
  - Autonomous
  - Data Center Chips
  - Data Center Systems

Plot comparing Accelerators

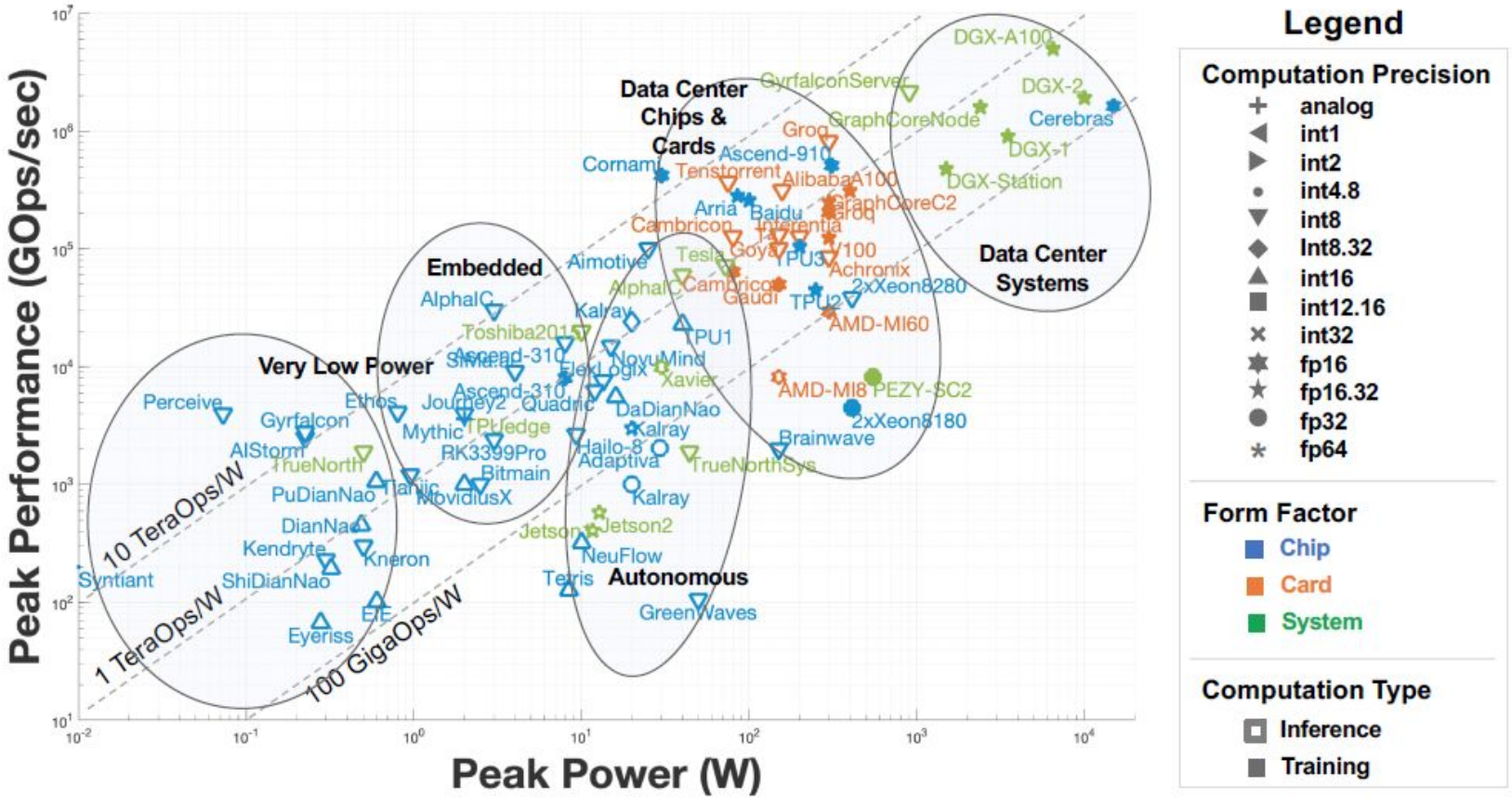


Fig. 2. Peak performance vs. power scatter plot of publicly announced AI accelerators and processors.

What affects the performance?

- Accelerators exploit the massive parallelization offered by matrix operations
- Data for such SIMD/MIMD architectures should be made available quickly
- The architectures of the execution units also must be optimized and pipelined to allow each unit to exploit ILP

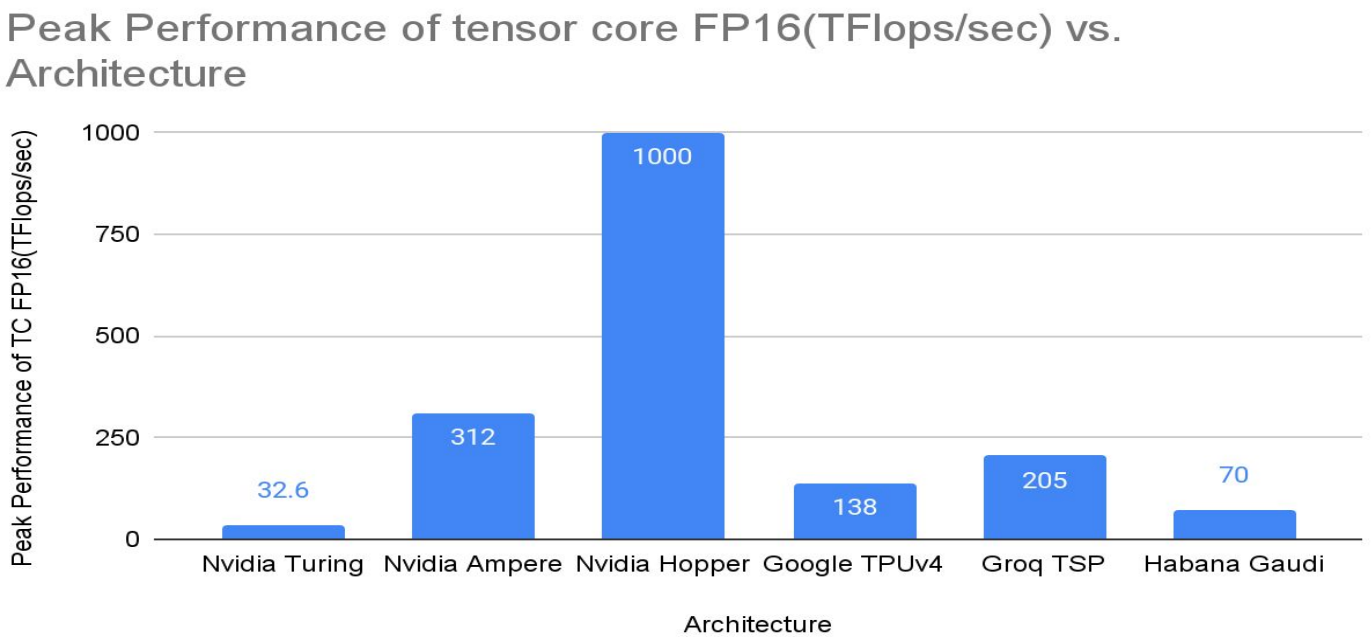
What affects the performance?

- Communication with the CPU should be seamless, and of large bandwidth
- Inter unit (core) communication

As accelerators must be compatible with most CPUs, the protocols used for CPU communication is a standard - PCIe Gen 4/5 and do not need to be compared

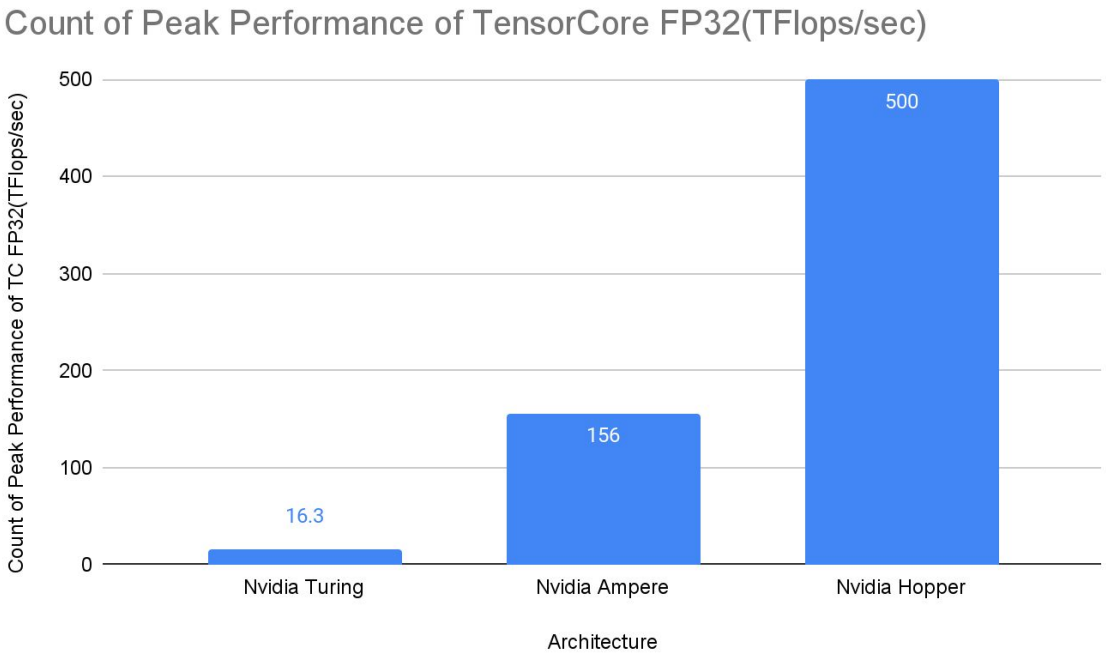
Comparisons

Performance

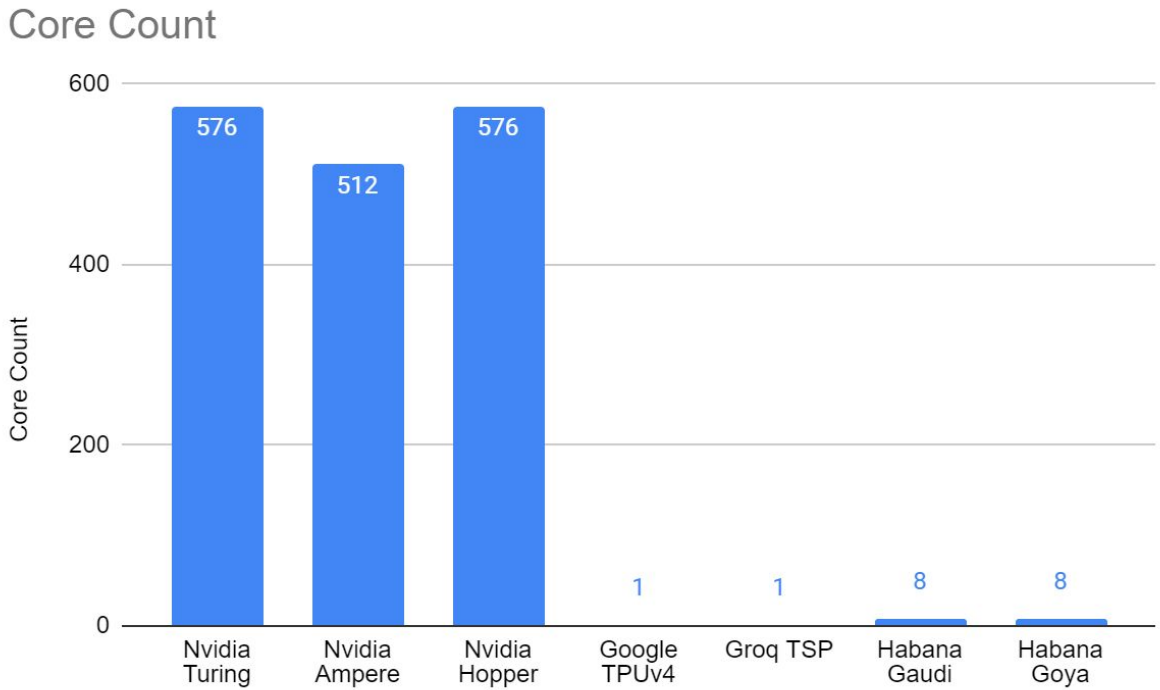


**Goya Reported a 2x performance benefit over T4 (Turing) and Gaudi a 7% increase over Goya for vision classification workloads*

Performance



Number of cores



Memory Organization

- Accelerators require large amounts of data to operate on SIMD workloads,
- Main Memory access being the bottleneck hinders performance
- Accelerators that have **larger cache sizes** and implement some form of **NUMA* memory architecture**, perform better

*Zoltan Majo and Thomas R. Gross. 2011. Memory management in NUMA multicore systems: trapped between cache contention and interconnect overhead. SIGPLAN Not. 46, 11 (November 2011), 11–20. https://doi.org/10.1145/2076022.1993481

Memory Organization - Hopper

- Hopper’s HBM3 delivers a 3 TB/sec of memory bandwidth
- 50 MB L2 cache architecture
- Uses a partitioned crossbar structure

** https://developer.nvidia.com/blog/nvidia-hopper-architecture-in-depth/*

Memory Organization - TSP

- The 88 SRAM slices provide the needed memory concurrency
- Each MEM slice comprises 20 tiles, arranged in a vertical stack, yielding a 2.5 Mibyte per-slice capacity
- The on-chip memory bandwidth is 27.5 TiB/s of SRAM bandwidth in each hemisphere
- No register file or cache, instead there is direct SRAM access

Dennis Abts et al. 2020. Think fast: a tensor streaming processor (TSP) for accelerating deep learning workloads. In Proceedings of the ACM/IEEE 47th Annual International Symposium on Computer Architecture (ISCA '20). IEEE Press, 145–158

Memory Organization - Ampere

- 192 KB of combined shared memory and L1 data cache.
- The A100 GPU includes 40 MB of L2 cache, divided into two partitions to enable higher bandwidth and lower latency
- The 40GB memory is organized as five active HBM2 stacks with eight memory dies perstack and delivers 1555 GB/sec

https://developer.nvidia.com/blog/nvidia-ampere-architecture-in-depth/

Memory Organization - TPUv4

- Delivers 614 GB/s bandwidth and 8GB off chip memory
- 16MB of on-chip memory in the form of VMEM, SMEM, and IMEM
- 128 MB Common Memory (CMEM) of TPUv4i acts as L2
- 512B native access size instead of 64B cache lines.
- the NUMA boundary is between parts inside the same core.

N. P. Jouppi et al., "Ten Lessons From Three Generations Shaped Google's TPUv4i : Industrial Product," 2021 ACM/IEEE 48th Annual International Symposium on Computer Architecture (ISCA), Valencia, Spain, 2021, pp. 1-14, doi: 10.1109/ISCA52012.2021.00010.

Memory Organization - Turing

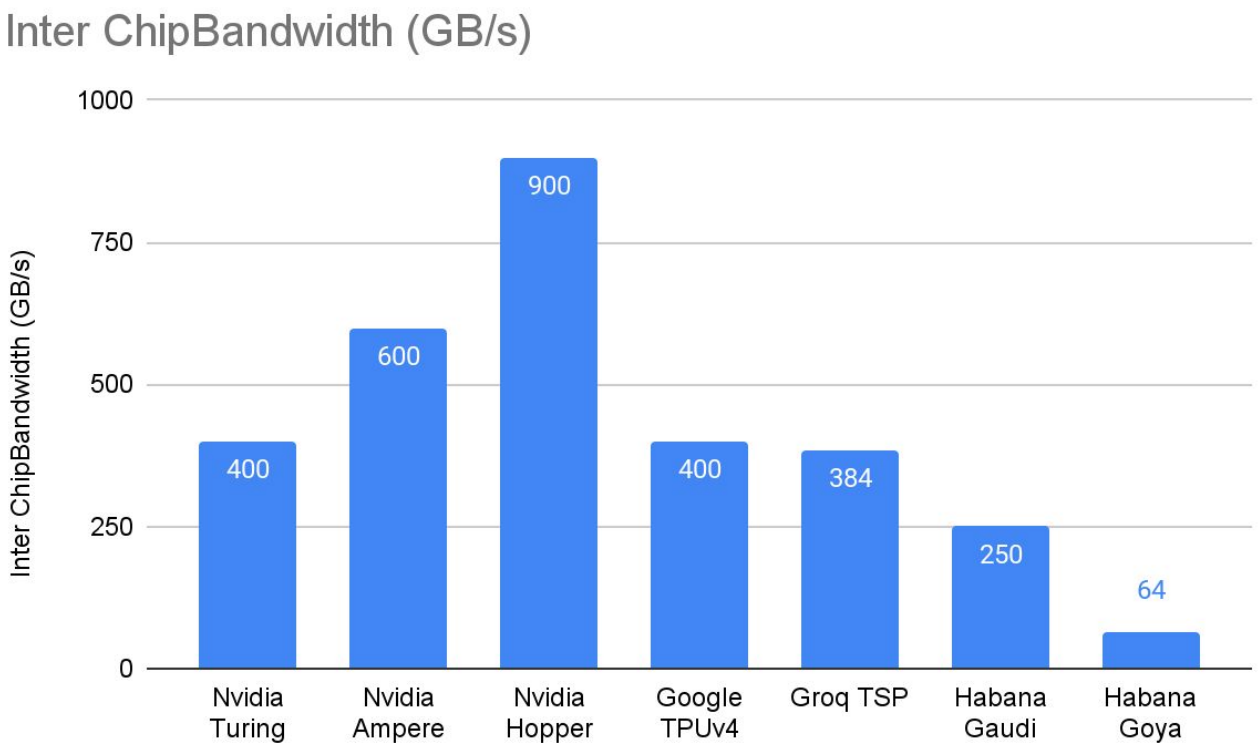
- SM is partitioned into four processing blocks, each with L0 instruction cache and a 64 KB register file
- The four processing blocks share a combined 96 KB L1 data cache/shared memory that can be reconfigured as 32/64 or 64/32
- 6 MB L2 cache
- Memory bandwidth of 672 GBps

https://developer.nvidia.com/blog/nvidia-turing-architecture-in-depth/

Interconnects

- Communication between multiple chips is important while running large workloads
- Faster communication enables data to move faster across chips effectively decreasing the compute time
- Accelerators performances benefit by having **faster interconnect speeds**

Interconnects



**Goya does not give an exact number, it has been taken from the technology used*

Interconnects

- Nvidia uses NVLink for communication with 50 Gbit/sec per signal pair
- Google’s TPU uses an on-device switch provides virtual-circuit, deadlock-free routing. To enable a 2D torus, the chip has four custom Inter-Core Interconnect (ICI) links, each running at 400Gbits/s per direction in TPUv4.

Interconnects

- The Gaudi processor embeds 20 pairs of 56-Gb/s Tx/Rx PAM4 serializers/deserializers (SerDes) that can be configured as 10 ports of 100 Gb Ethernet, 20 ports of 50-Gb/25-Gb Ethernet
- Goya uses PCIe Gen416 enabling communication to any host of choice, FPGA or peer-to-peer communication with another Goya

Architecture

- The micro-architecture plays the most important role in determining the performance of accelerators
- This involves design of **custom hardware** that exploits parallelism to increase the throughput

Architecture - Turing

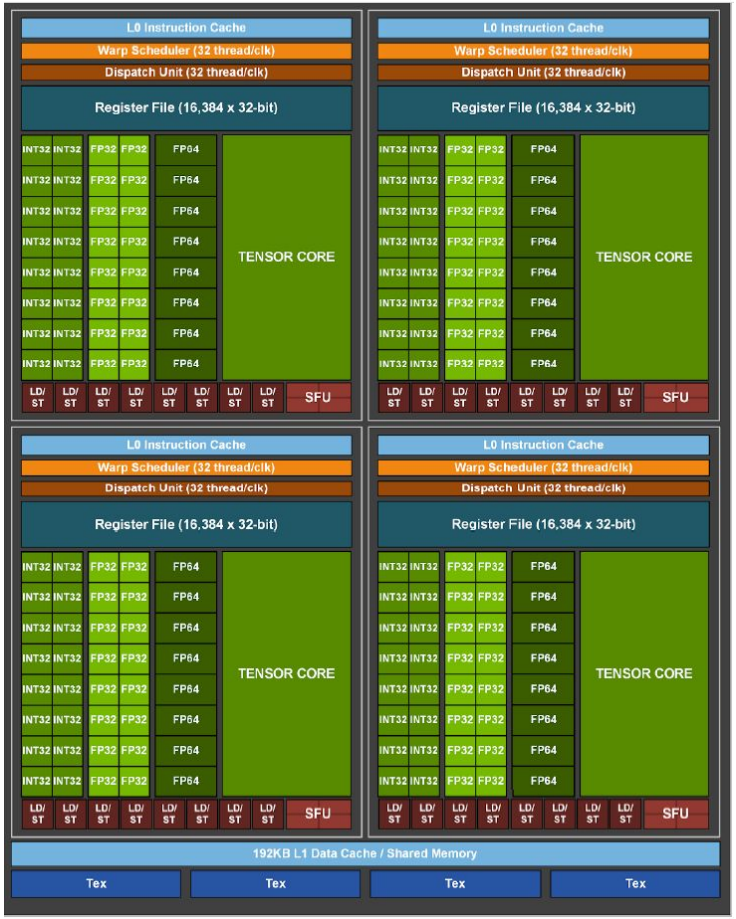
- 6 GPCs
- 36 TPCs (6 TPCs/GPC)
- 2 SMs/TPC
- 72 SMs per full GPU

Architecture - Turing

- Utilize lossless memory compression techniques to reduce memory bandwidth demands
- This reduces the amount of data written out to memory and transferred from memory to the L2 cache
- Introduced Turing Tensor Cores that target matrix multiplication

Architecture - Ampere

- 8 GPCs
- 8 TPCs/GPC
- 2 SMs/TPC
- 16 SMs/GPC
- 128 SMs per full GPU



Architecture - Ampere

- The third gen Tensor Cores for high-performance matrix multiply and accumulate (MMA)
- Features asynchronous data transfer from main memory to shared memory
- Exploiting matrix sparsity it doubles compute throughput for deep neural networks

\

Architecture - Ampere

- It can divide a single GPU into multiple GPU partitions called GPU instances, enabling multiple GPU Instances to run in parallel on a single, physical A100 GPU (MIG feature)

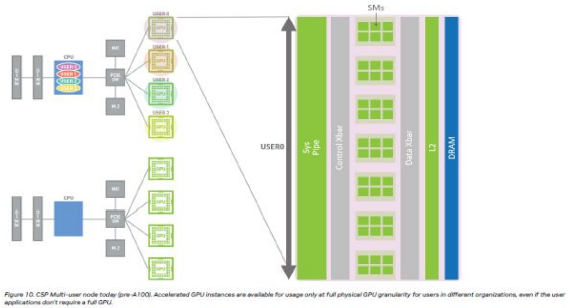


Figure 10. CSP Multi-user node today (pre-A100). Accelerated GPU instances are available for usage only at full physical GPU granularity for users in different organizations, even if the user applications don't require a full GPU.

CSP Multi-Instance GPU (MIG)

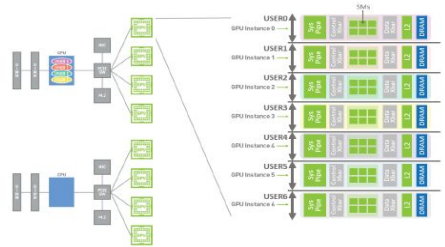


Figure 11. CSP multi-user with MIG diagram. Multiple independent users from the same or different organizations can be assigned their own dedicated, protected, and isolated GPU instance within a single physical GPU.

- 8 GPCs
- 72 TPCs (9 TPCs/GPC)
- 2 SMs/TPC
- 144 SMs per full GPU

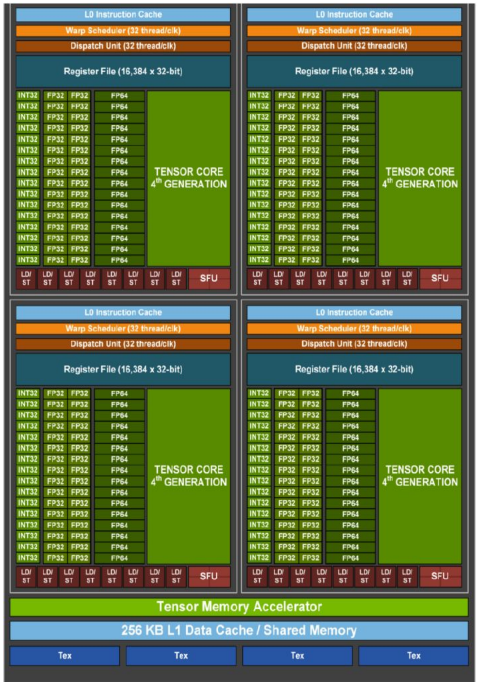


Figure showing 1 SM

Architecture - Hopper

- The fourth gen Tensor Cores for high-performance matrix multiply and accumulate (MMA) operations perform 2x faster than Ampere
- Feature added to exploit matrix sparsity to further accelerate performance
- Expanding ISA
- features asynchronous execution

\

Architecture - Hopper

- The tensor memory accelerator allows tensor dimensions and block coordinates instead of per-element addressing
- Large blocks up to the shared memory capacity can be specified and loaded from global memory into shared memory and vice versa

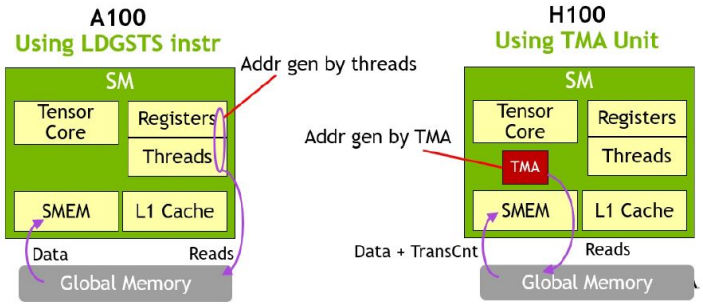
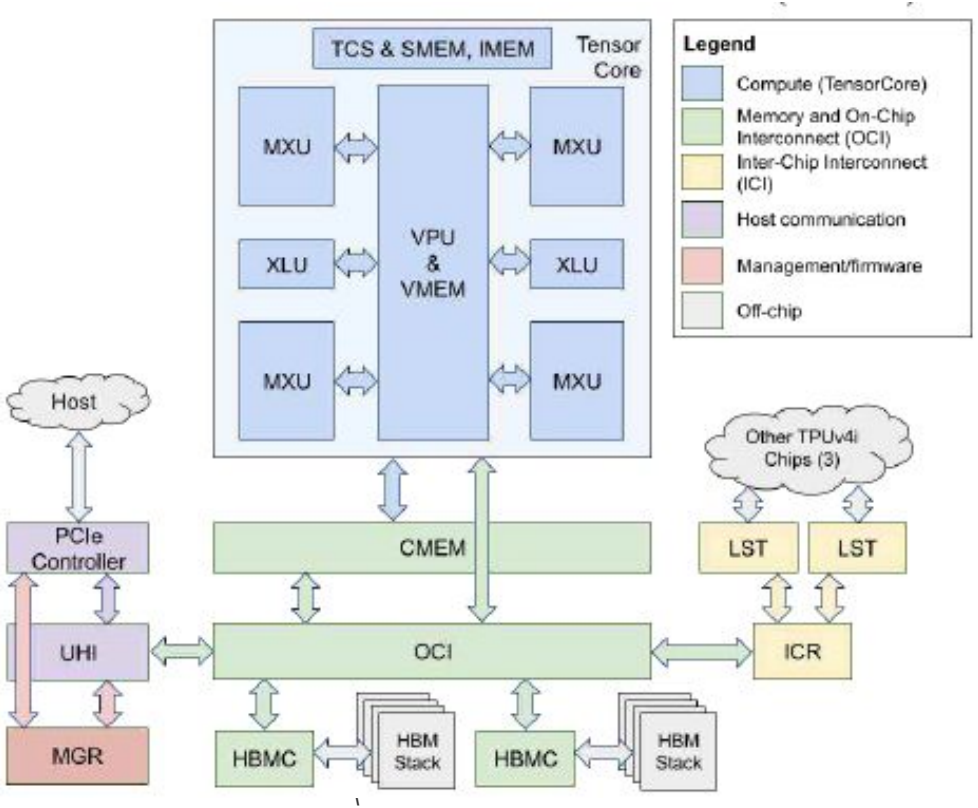


Figure 16. Asynchronous memory copy with TMA on H100 vs. LDGSTS on A100

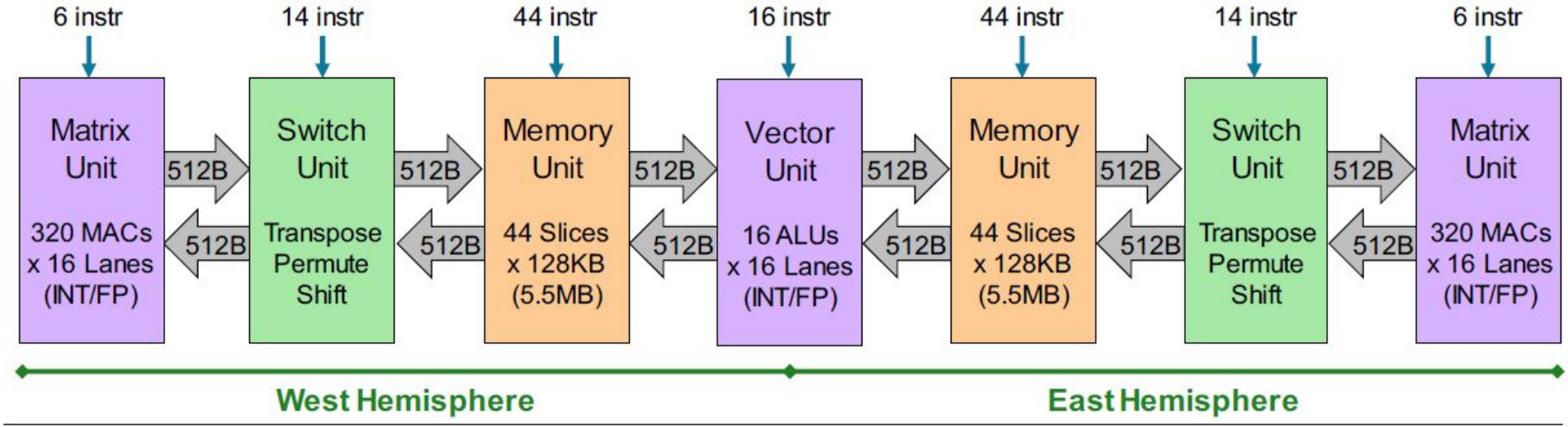
Architecture - TPU



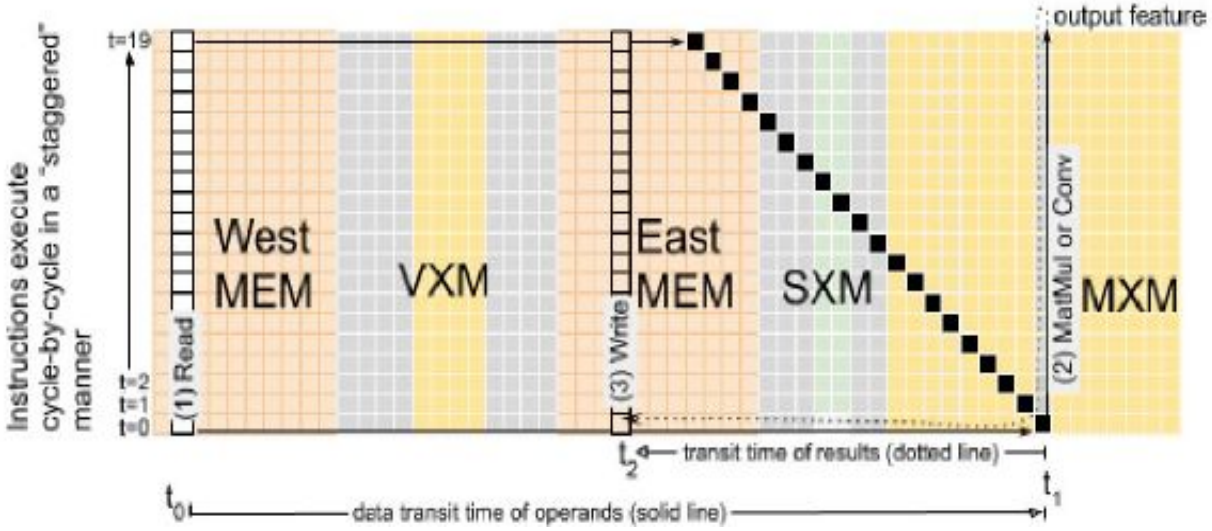
Architecture - TPU

- Software managed Imem
- VLIW instructions: 322 bits can launch eight operations:2 scalar, 2 vector ALU, vector load and store, and a pair of slots that queue data to and from the matrix
- VPU performs vector operations using a large on-chip vector memory 2D vector registers (Vregs)
- The VPU streams data to and from the MXU

Architecture - TSP

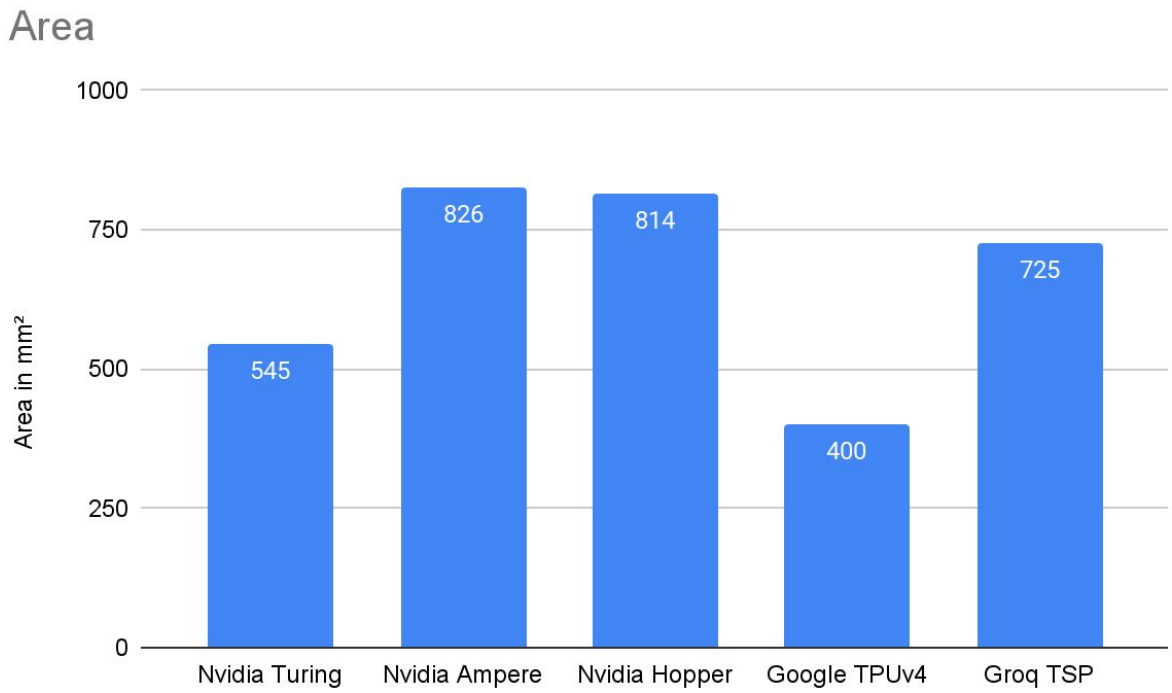


Architecture - TSP

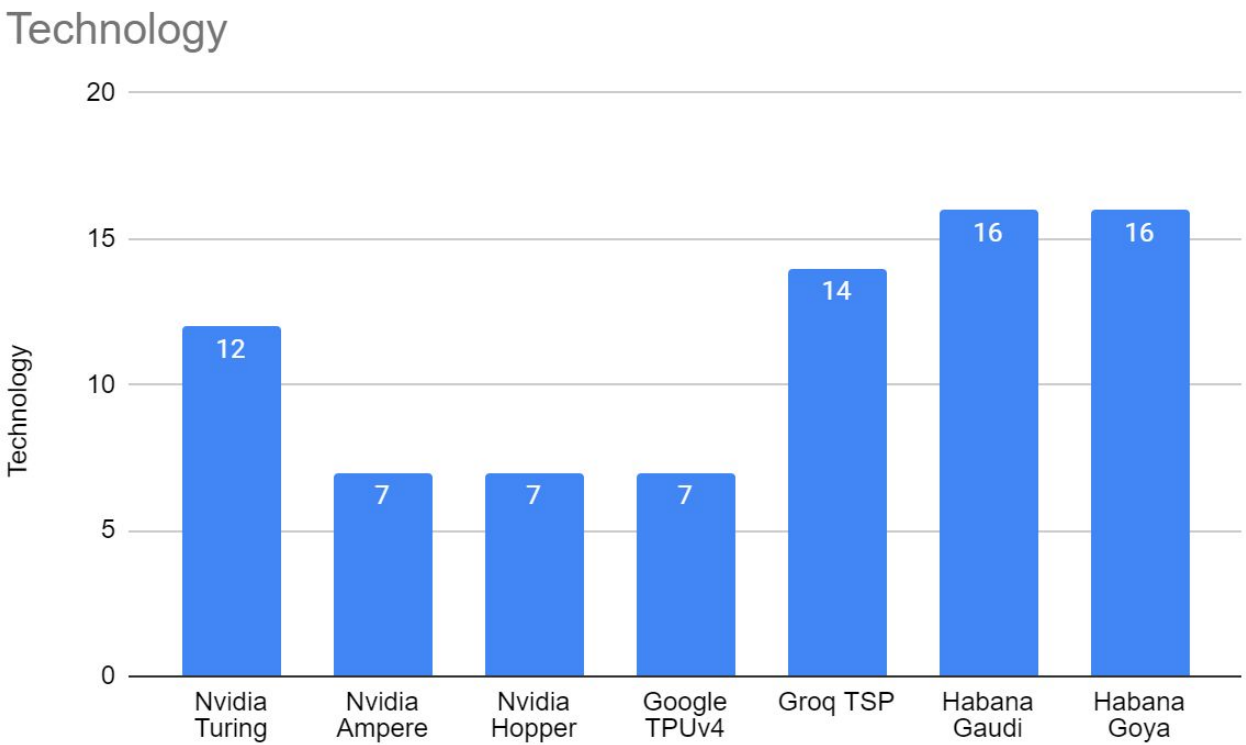


Instruction Control						
Matrix	S	Memory	V	Memory	S	Matrix
Matrix	S	Memory	V	Memory	S	Matrix
Matrix	S	Memory	V	Memory	S	Matrix
Matrix	S	Memory	V	Memory	S	Matrix
Matrix	S	Memory	V	Memory	S	Matrix
Matrix	S	Memory	V	Memory	S	Matrix
Matrix	S	Memory	V	Memory	S	Matrix
Matrix	S	Memory	V	Memory	S	Matrix
Matrix	S	Memory	V	Memory	S	Matrix
Matrix	S	Memory	V	Memory	S	Matrix
Matrix	S	Memory	V	Memory	S	Matrix
Matrix	S	Memory	V	Memory	S	Matrix
Matrix	S	Memory	V	Memory	S	Matrix
Matrix	S	Memory	V	Memory	S	Matrix
Matrix	S	Memory	V	Memory	S	Matrix
Matrix	S	Memory	V	Memory	S	Matrix
Matrix	S	Memory	V	Memory	S	Matrix
Matrix	S	Memory	V	Memory	S	Matrix
Matrix	S	Memory	V	Memory	S	Matrix
Matrix	S	Memory	V	Memory	S	Matrix
Matrix	S	Memory	V	Memory	S	Matrix
PCIe and Other I/O						

Area


Area
1000
750
500
250
0
Area in mm²
545
826
814
400
725
Nvidia Turing
Nvidia Ampere
Nvidia Hopper
Google TPUv4
Groq TSP

Technology


Technology
Technology
20
15
10
5
0
12
7
7
7
14
16
16
Nvidia Turing
Nvidia Ampere
Nvidia Hopper
Google TPUv4
Groq TSP
Habana Gaudi
Habana Goya

Inference

- The main aim of hardware accelerators is to exploit parallelizability of code
- Priority is given to performance over cost and power
- Power is considered to be of higher priority than the cost due to deployability concerns

\

Takeaways

- Nvidia have progressively packed more cores/GPU to squeeze more TFlops
  - This was possible because of instruction independence in SIMD

\

Conclusions

- **Having custom hardware** units for improving computational speed has the best potential for increasing performance
- Accelerators performances benefit by having **faster interconnect speeds**
- Accelerators that have **larger cache sizes** and implement some form of **memory slicing**, perform better

Why are there so many GPUs?

Scheduling and Execution

- GPU design analogous to CPU design targets the instruction scheduling and execution stage of the pipeline
- Instruction scheduling is an NP-hard problem*, it is challenging to derive the perfect scheduling order without oracle knowledge
- Execution too does not have just one way to maximize throughput

*David Bernstein, Michael Rodeh, and Izidor Gertner. 1989. On the complexity of scheduling problems for parallel/pipelined machines. IEEE Transactions on computers 38, 9 (1989), 1308–1313.

\

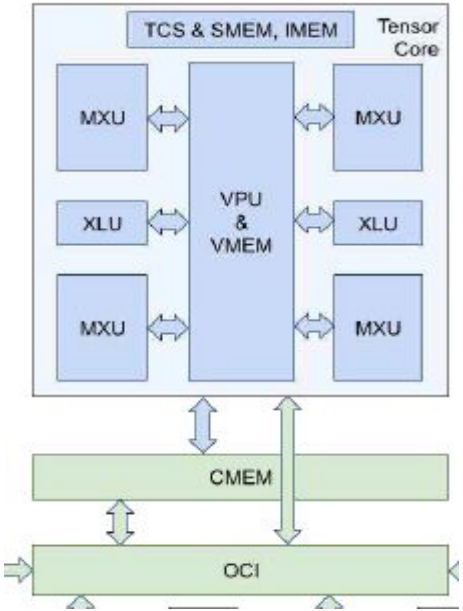
- Competition rises in the unique ways used to tackle instruction scheduling and execution constrained to maximizing throughput
- There being no ‘correct’ way has led to numerous GPU architectures

- The different stages of GPU execution such IF, ID, Exec, WB are very similar
- The Exec stage can have different logical implementations, but has identical functional units
- Differences show up only in the manner of instruction scheduling

Instruction Schedule Stage comparison

Instruction Scheduling

- Nvidia’s GPU schedule instruction to cores in round robin
- Groq’s TSP schedules instruction based on locality of operands
- TPU’s Core Sequencer fetches from the Imem, executes scalar operations and forwards vector instructions to the VPU. This can launch eight operations



TPU v4

- Nvidia’s Round Robin although simple, is ideal for the large number of cores present
- Groq’s TSP scheduler is unique but complex; it overcompensates for the single core it is

Advantages and disadvantages

Advantages and Disadvantages

Architecture	Advantages	Disadvantages
Nvidia Hopper	Highest performance, Scalable Architecture, Is compiler independant	Enormous power consumption, Expensive
Groq TSP	Highest single core performance, Low power consumption	Not scalable, complicated & dedicated compiler design, cannot use any compiler
Google TPU	Compiler Independent, Scalable	Low performance

References

1. Zoltan Majo and Thomas R. Gross. 2011. Memory management in NUMA multicore systems: trapped between cache contention and interconnect overhead. SIGPLAN Not. 46, 11 (November 2011), 11–20. https://doi.org/10.1145/2076022.1993481
2. https://developer.nvidia.com/blog/nvidia-turing-architecture-in-depth/
3. https://developer.nvidia.com/blog/nvidia-ampere-architecture-in-depth/
4. https://developer.nvidia.com/blog/nvidia-hopper-architecture-in-depth/
5. Dennis Abts et al. 2020. Think fast: a tensor streaming processor (TSP) for accelerating deep learning workloads. In Proceedings of the ACM/IEEE 47th Annual International Symposium on Computer Architecture (ISCA '20). IEEE Press, 145–158
6. Norman P. Jouppi, Doe Hyun Yoon, George Kurian, Sheng Li, Nishant Patil, James Laudon, Cliff Young, and David Patterson. 2020. A domain-specific supercomputer for training deep neural networks. Commun. ACM 63, 7 (July 2020), 67–78.
7. N. P. Jouppi et al., "Ten Lessons From Three Generations Shaped Google's TPUv4i : Industrial Product," 2021 ACM/IEEE 48th Annual International Symposium on Computer Architecture (ISCA), Valencia, Spain, 2021, pp. 1-14, doi: 10.1109/ISCA52012.2021.00010.
8. A. Reuther, P. Michaleas, M. Jones, V. Gadepally, S. Samsi and J. Kepner, "Survey of Machine Learning Accelerators," 2020 IEEE High Performance Extreme Computing Conference (HPEC), Waltham, MA, USA, 2020, pp. 1-12, doi: 10.1109/HPEC43674.2020.9286149.
9. E. Medina and E. Dagan, "Habana Labs Purpose-Built AI Inference and Training Processor Architectures: Scaling AI Training Systems Using Standard Ethernet With Gaudi Processor," in IEEE Micro, vol. 40, no. 2, pp. 17-24, 1 March-April 2020, doi: 10.1109/MM.2020.2975185.
10. https://grog.com/wp-content/uploads/2023/05/GROQ-ROCKS-NEURAL-NETWORKS.pdf
11. Reuther, Albert, et al. "AI and ML Accelerator Survey and Trends." 2022 IEEE High Performance Extreme Computing Conference (HPEC), Sept. 2022. Crossref,
12. Tor M. Aamodt, Wilson Wai Lun Fung, and Timothy G. Rogers. 2018. General-purpose Graphics Processor Architectures. Morgan & Claypool Publishers.
13. E. Rotem et al., "Intel Alder Lake CPU Architectures," in IEEE Micro, vol. 42, no. 3, pp. 13-19, 1 May-June 2022, doi: 10.1109/MM.2022.3164338.
14. https://hc34.hotchips.org/assets/program/conference/day2/Machine%20Learning/HotChips34%20-%20Grog%20-%20Abts%20-%20final.pdf
15. https://hc32.hotchips.org/assets/program/conference/day2/HotChips2020_ML_Training_Google_Norrie_Patil.v01.pdf
16. https://hc34.hotchips.org/assets/program/conference/day1/GPU%20HPC/HC2022.NVIDIA.Choquette.vfinal01.pdf

Thank You