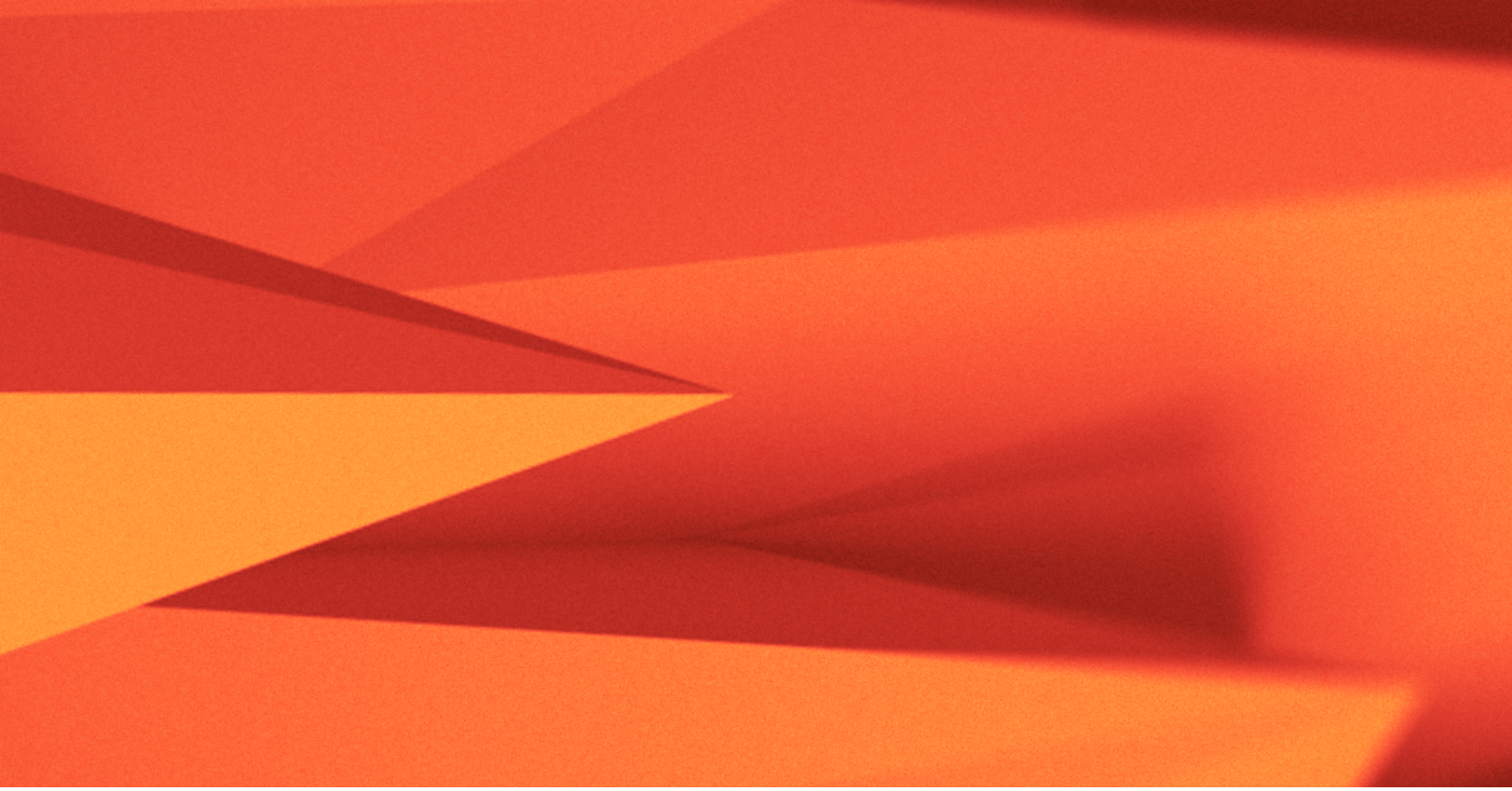

Delivering Low Latency for Real-time AI and HPC.

Your inference - right now.

Author: Groq Inc.

Date: Q4 2021
Version: Delivering Low Latency for Real-time AI and HPC v0.1

Legal Statements

Delivering Low Latency for Real-time AI and HPC

Groq Tech Doc

While inference applications are becoming increasingly complex, the ability to operate within the real world in natural harmony with humans is non-trivial. Generally, classes of low latency inference applications are driven by at least two needs. First there is the need to interact with humans using, by way of example, natural language processing for Q&A, translations, pair coding and similar applications. Second, in other applications there is the need to integrate with a control or feedback system in the real world, like autonomous vehicles, industrial control, or even fusion reactors where low latency can really matter.

Because of Moore's law, which says transistor density doubles roughly every two years, a typical semiconductor device can double compute density. But critically, memory bandwidth and memory latency has not scaled at the same pace as compute density. This decoupling of core technological capabilities has forced architects to make trade-offs that enable the extra compute capacity to be used while interfacing with a much more constrained memory architecture. To some degree this includes architectural enhancements like caching, but fundamentally it has changed the way that compute is done. For inference applications, this has meant increasing the batch size.

Increased batch size to overcome memory bandwidth limitations

The workhorse operations behind machine learning inference are usually vector-matrix and matrix-matrix operations which consume massive amounts of data. For matrix-matrix operations, the compute density of a matmul is $O(N^3)$ for an NxN matrix with memory requirements for weights of N^2.

Forgoing a detailed analysis, we can look at the systolic structure with N^2 components with the assumption of a compute time of N cycles . Of course, this assumes the ability to load N^2 weights in that time. When using most memory architectures, this results in a clear bottleneck, and most systems today work around this by minimizing the loading of weights by amortizing that cost across multiple batches of compute.

This is the case for GPUs which were historically designed for massively parallel processing that takes place in hundreds or thousands of cores concurrently, and built on multi-task fixed-structure processing engines. The parallelism is dependent on a pipeline architecture with each processing stage adding new data to ongoing parallel calculations.

Figure 1 shows this process on a traditional GPU. As processing units make references to the local cache, it begins the cumbersome back and forth of cache fills and write-backs. These processing units are part of a streaming multiprocessor core in which all cores are dynamically contending for

the shared DRAM controller. Particularly, all 80 cores are executing thread warps that implement a total of 32 single instruction multiple threads, for a grand total of 2560 threads contending for 16 pseudo channels of HBM2e in the shared DRAM controller. This results in long wait times and contributes to non-determinism at the system level because of dynamic contention for a shared resource that causes reordering, latency variance, and unpredictable performance.

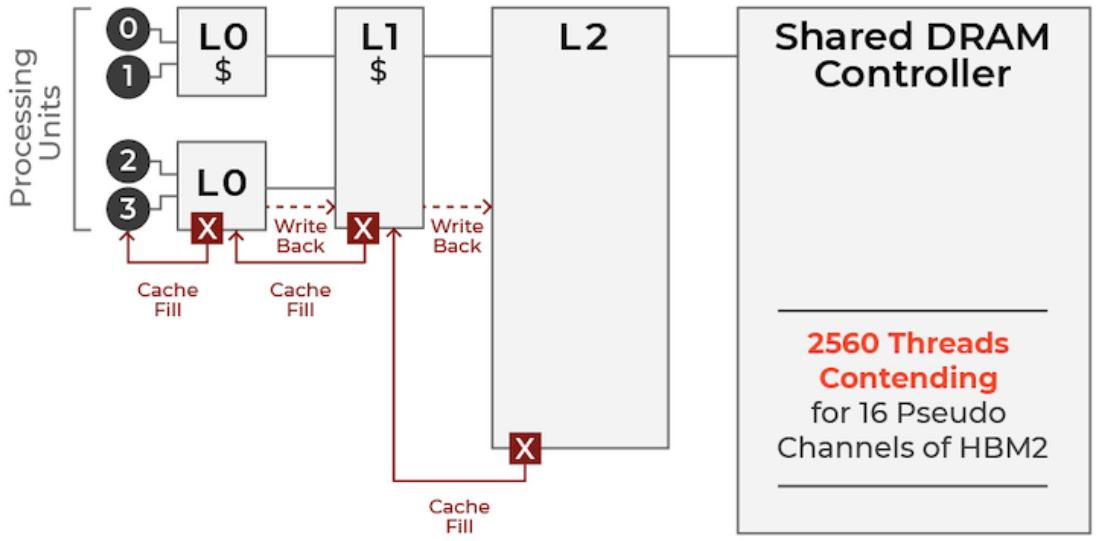

Figure 1: Traditional GPU Memory Hierarchy

When used in machine learning applications, GPUs experience significant declines in performance when fed data in small batch sizes, because continuous weight reloads introduce stalls in GPU execution. This latency can introduce a dramatic impact on real-time inference performance at batch size 1, like in the examples above, meaning GPU-based platforms may not deliver inference performance fast enough. Furthermore, it impacts TCO for machine learning platform investments.

These factors have created an unmet market need for high performance at batch 1 and this is why Groq chose it as a key tenant of its Tensor Streaming Processor architecture. The GroqChip™ processor is purpose-built to service latency-dependent workloads.

Figure 2 shows how GroqChip exploits heterogeneous threads or “straight line” threads, meaning threads do not branch but instead correspond to the different functional units for execution. Data pathways are fully pipelined in both directions. Instructions are pipelined vertically while data streams flow east and west, intersecting the functional units to perform operations, taking advantage of locality. We can read a quantity from memory, operate on that quantity on the vector unit, and store the results back to memory. Additionally, GroqChip decouples compute and memory access - this is critical for higher memory level parallelism, enabling many reads and

writes in flight. This means GroqChip computes and communicates effectively in a single step, delivering low latency, high performance, and accuracy with predictability.

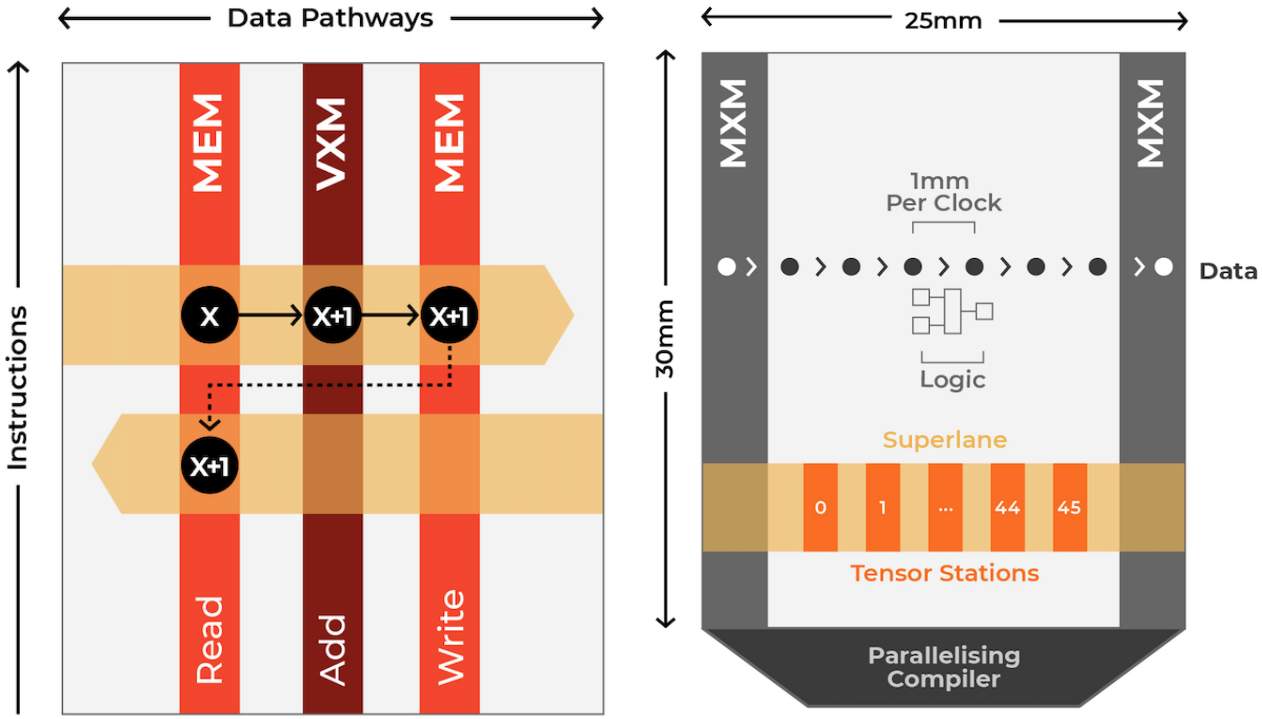

Figure 2 & Figure 3: GroqChip™ processor memory architecture

Groq's architecture is the opposite of high batch GPUs with the GroqChip processor having 230 MB of SRAM delivering 80TB/s of on-chip bandwidth. Figure 3 shows how GroqChip exposes instruction level parallelism, memory level parallelism, and data level parallelism very efficiently, taking the unique approach of computing and communicating simultaneously. After it's development, control was turned over to the software side to build a large scale, parallelising compiler to take advantage of all these forms of concurrency. This helps contribute to Groq's ability to provide high performance at batch 1. A 256 batch for training is typical in other architectures, meaning 256 images must be processed and "learned from '' before an application can provide information about the first one. With Groq operating at batch 1 and thus processing each image as it's received (as opposed to waiting for all 256), not only does waiting decrease, accuracy increases. Plus, the Groq architecture allows developers to not amortize long latencies inherent in GPUs and other traditional architectures.

Bottom line: it's fast, accurate, and the computation uses less energy to move data, yielding unparalleled efficiencies.

Three Points That Matter

- An architecture built for real-time AI, inherently capable of batch 1
- Batch 1 algorithmic agility

- Faster weight loading means predictable low latency

Use Cases

Autonomous Vehicles

Within Advanced driver-assistance systems (ADAS), capabilities like light detection and ranging (LIDAR), cameras and sensors in a vehicle need to be able to instantly classify, identify, and react based upon observations. These edge devices can use object detection and behavioral prediction models to measure not just that there's a pedestrian on the corner, but the angle of their body, the tilt of their head, the direction of their eyes and the movement of their legs—all to determine if they're going to take that step off the curb. This is a perfect and very important job for batch 1 because waiting to process additional frames to determine what was observed in the first one could be fatal. With batch 1, every frame is analyzed and acted on immediately. So while GPUs and other architectures may be able to match the volume of compute, it requires large batch sizes and additional time, often to the detriment of accuracy. The wait time alone that this comes with is a non-starter for autonomous.

Search Engine Queries

Search engines are required to crawl not only web sites, but videos, images, infographics, books, whitepapers, databases, and directories. A search engine needs to accept and recognize a query, determine the words and their order, look up the information, edit results based on filter bubbles (one's activity history), rank it all, and finally send it back to the user. Millions of results—in less than a second. The two most important factors of these results are speed and accuracy. Which is why batch 1 is the perfect solution. Since every single query is analyzed as it comes, it's much faster—there's no waiting for additional queries before delivering a result to a processor. And since only one query is being analyzed, it's much more accurate.

By The Numbers

High Throughput and Low Latency at Batch 1 on ResNet 50

Groq implemented ResNet 50, an inference benchmark that measures machine learning accelerator performance. ResNet-50 typically sees faster performance with large batch sizes, yet Groq's batch 1 results demonstrated outstanding performance and latency.

Groq's batch 1 results clocked 18,800 inferences per second with a latency of only ~0.05msec

- The latest gen GPU competition was 4,197 inferences per second at batch 1, trailing Groq's ~4.5x better results[1]
- GPUs could compete with throughput, but only at much higher batch sizes

1As measured by comparing ResNet 50 batch 1 results for GroqCard™ accelerator vs Nvidia's published A100-PCIE-40GB results on nivdia.com.

Deterministic machine learning for latency critical control systems at Argonne National Laboratory

In working with Argonne National Laboratory, the goal was to maximize performance within a 1ms hard requirement needed for forecasting plasma instabilities in tokamak reactors. Groq exceeded this requirement at 193k IPS at 0.6ms, giving 622x higher throughput compared to the lowest latency GPU result as shown in Figure 4.

Groq's advantage lies in the fact that the reactor's control system requires predictable execution within a specified time window for operational efficiency and safety. Groq's deterministic architecture provided zero variation in time to result. Additionally, with such high performance at low batch sizes, this enabled more complex algorithms within the required response time window.

Ultimately, Groq's architecture delivers deterministic machine learning at low latency, with 5x-20x performance while meeting tight response window requirements.[2]

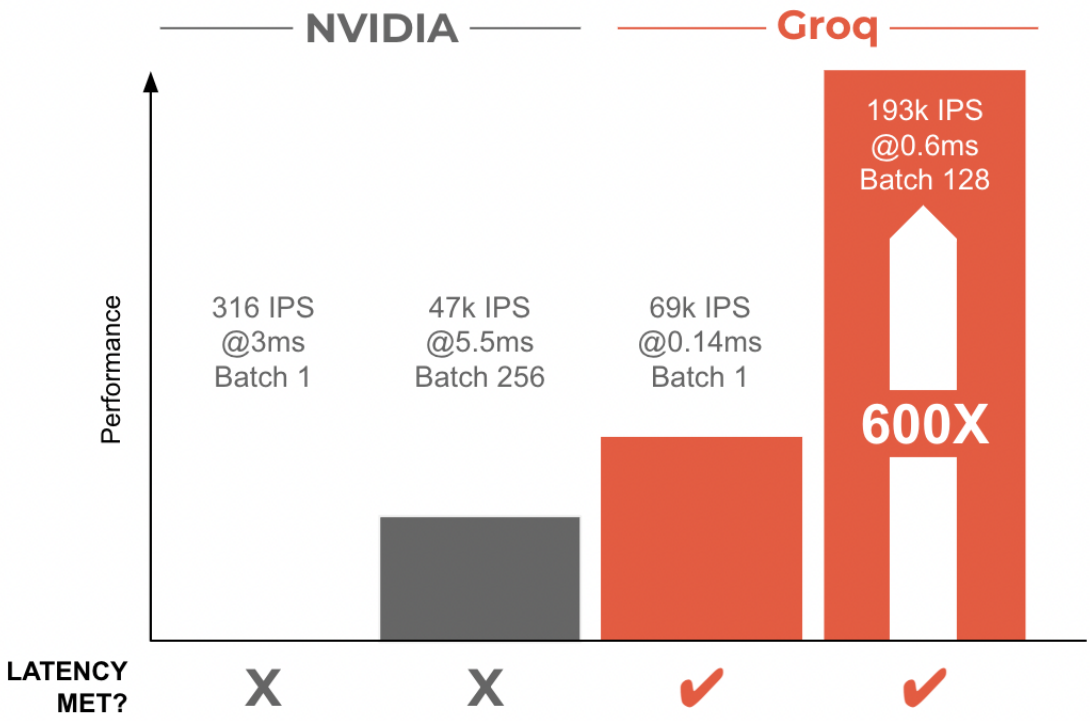

Figure 4: Performance comparison for latency critical control systems at Argonne National Laboratory. **Latest Gen HPC GPU (Nvidia A100):** 316 IPS @ 3ms, Batch 1, FP32; 46k IPS @5.5ms Batch 256, FP32. **GroqCard™ accelerator:** 69k IPS @0.14ms, Batch 1, TruePoint™ 16; 193k IPS @0.6ms, Batch 128, TruePoint™ 16.

2 As measured by comparing IPS at the fastest possible latency for both accelerators (GroqCard™ 1 vs Nvidia A100)

Interested in Learning More?

For more information on Groq technology and products, contact us at info@groq.com, follow us on Twitter @GroqInc and connect with us on LinkedIn https://www.linkedin.com/company/groq.

Groq Inc. HQ
400 Castro St. Suite 600
Mountain View, CA 94041

Mailing Address
PO Box 1778
Mountain View, CA 94041

www.groq.com
support.groq.com