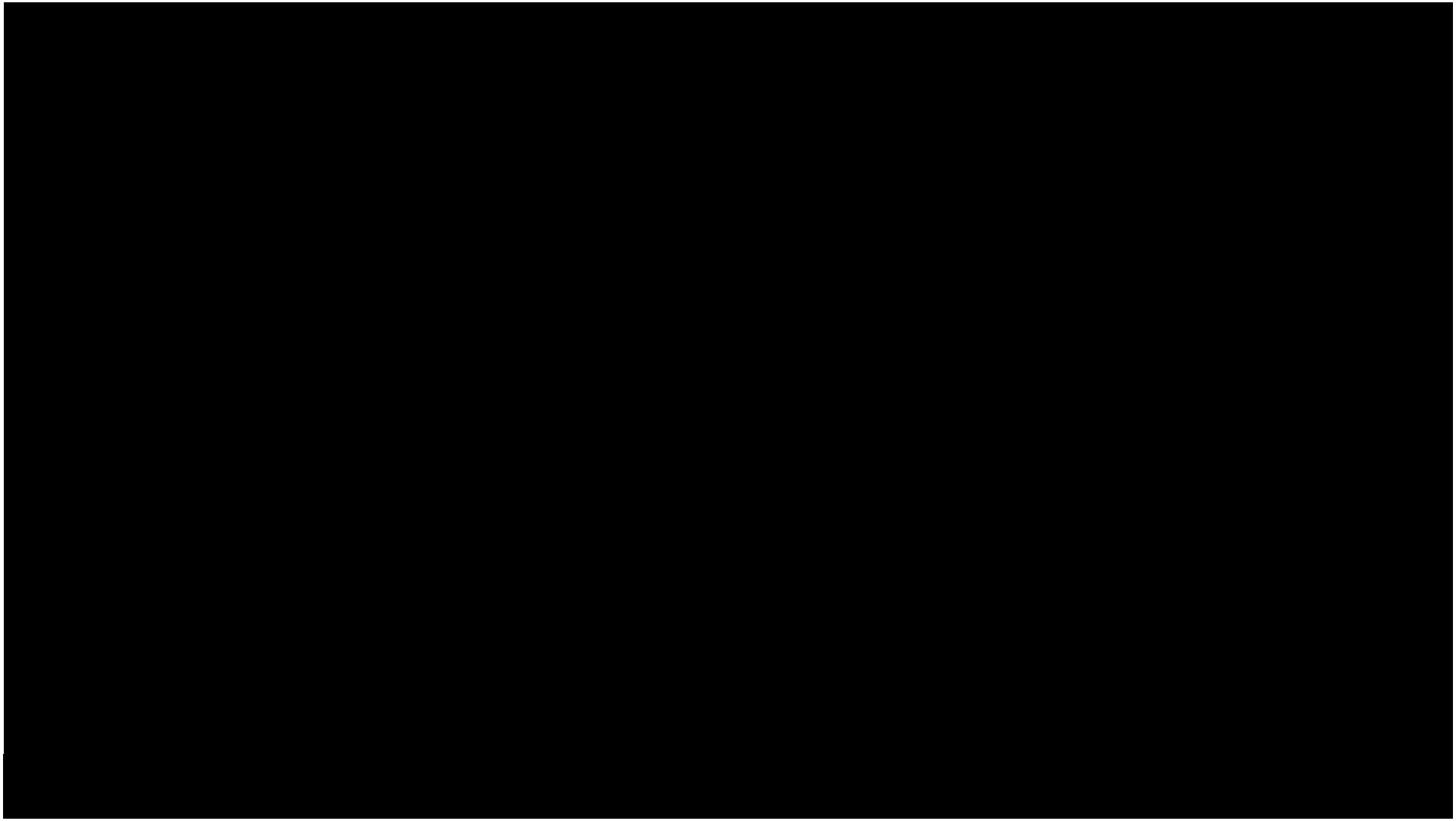
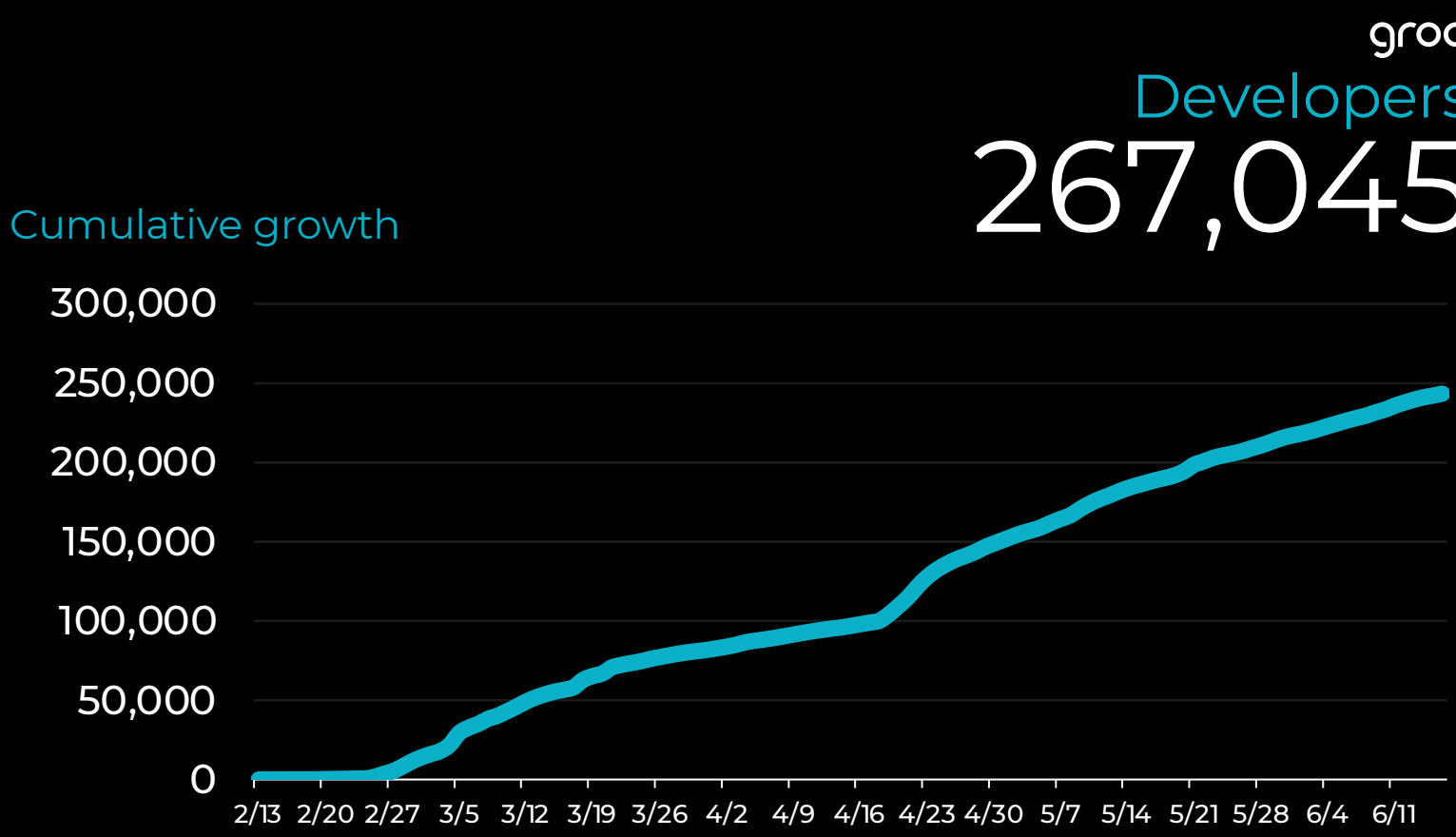

All-In Podcast



groq

Developers

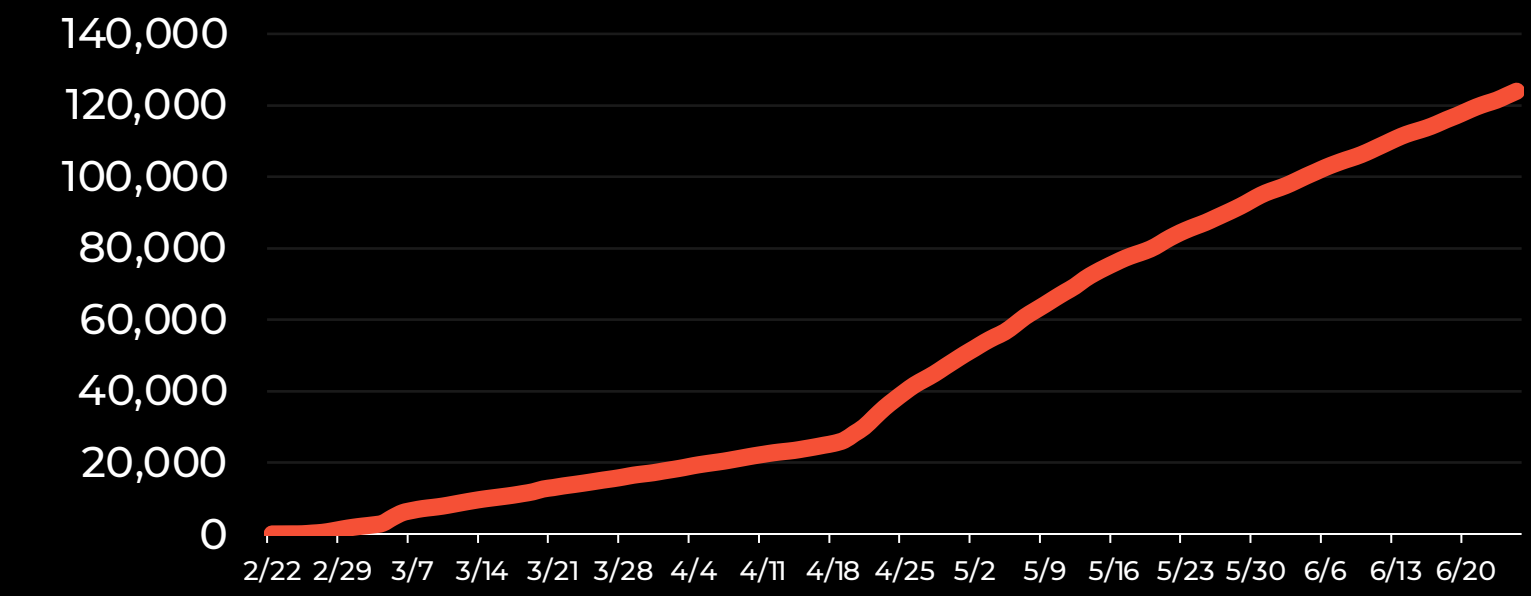
267,045



groq

Active API Keys

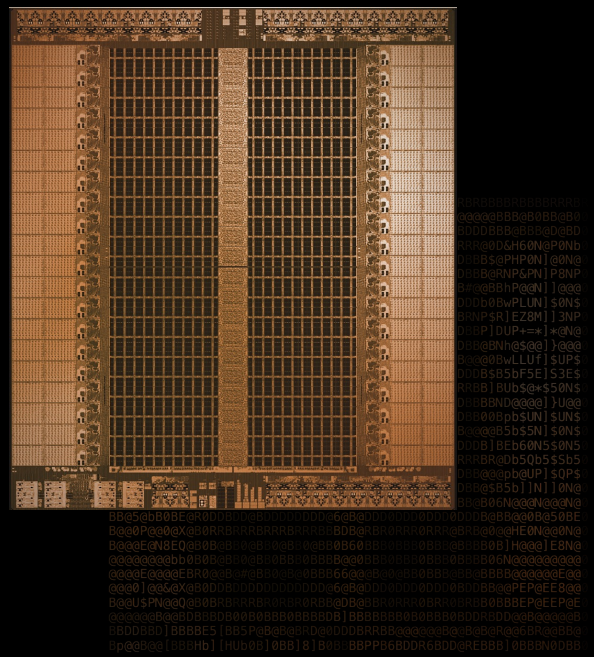
128,026



Igor Arsovski

Chief Architect, Fellow

groq®

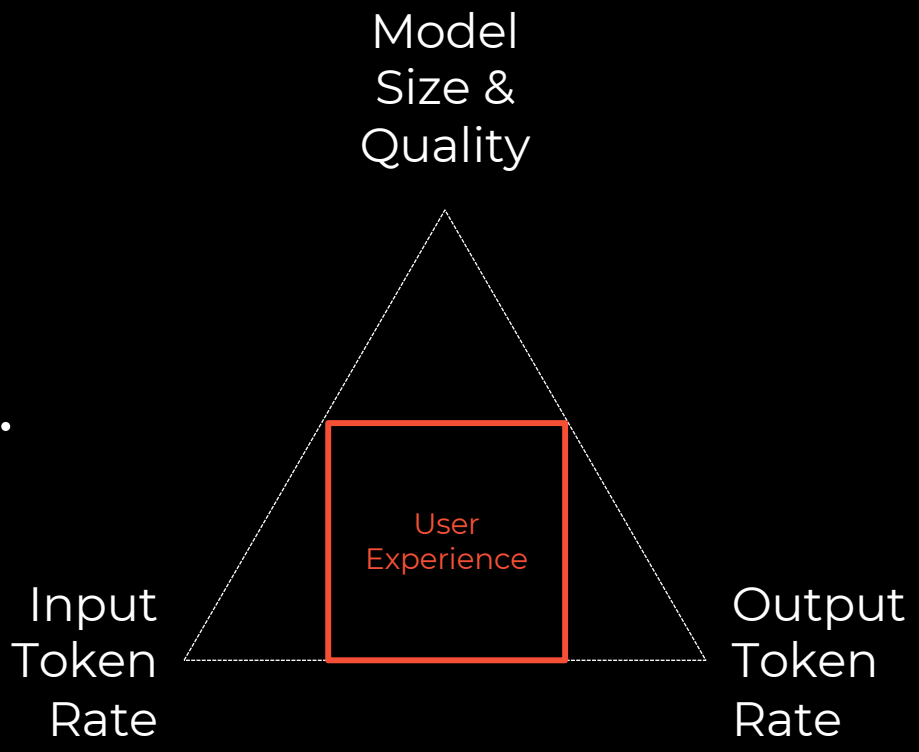


Fast AI Inference.

6x	4x	⅓
Faster	Cheaper	Energy
	14nm vs 4nm	

groq

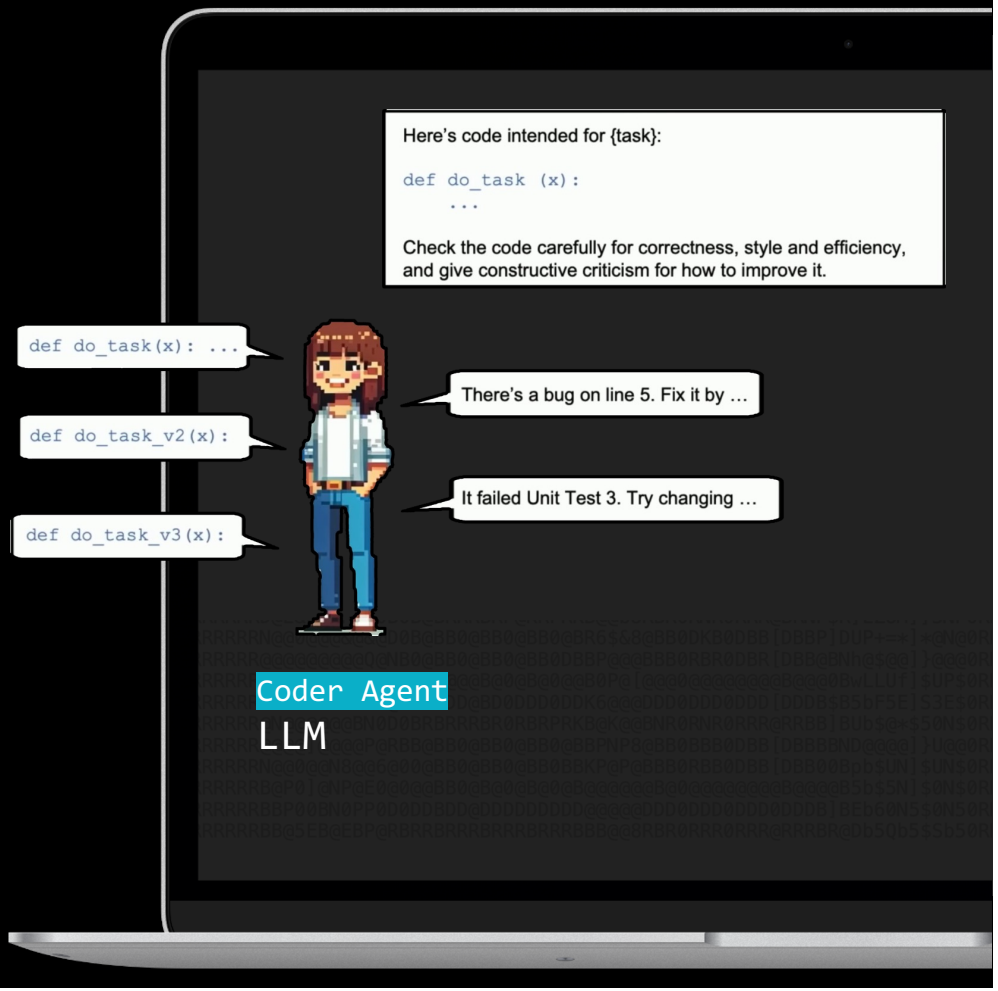
Fast AI Inference.



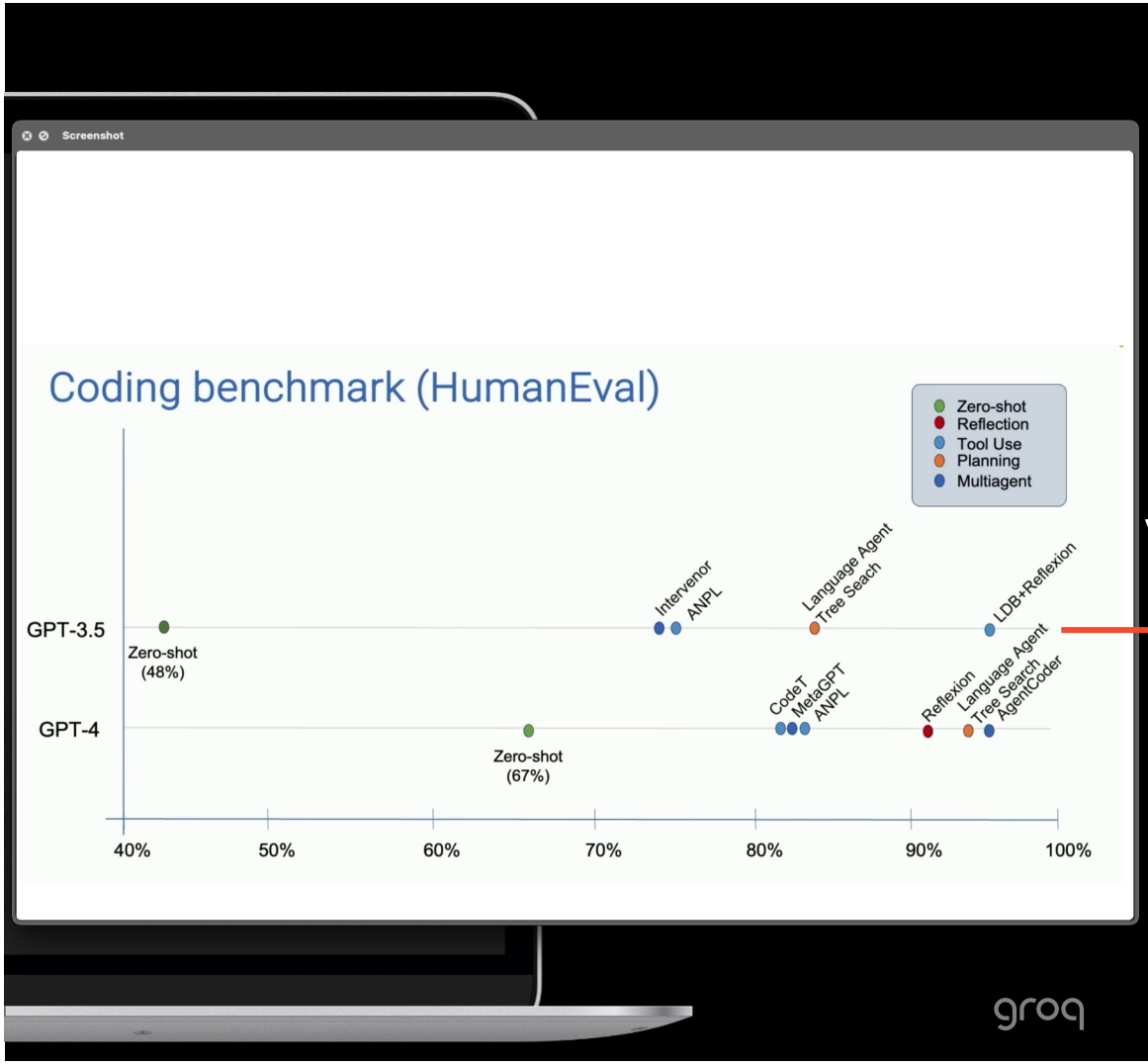
groq

Reflection.

"Please write code for {task}"



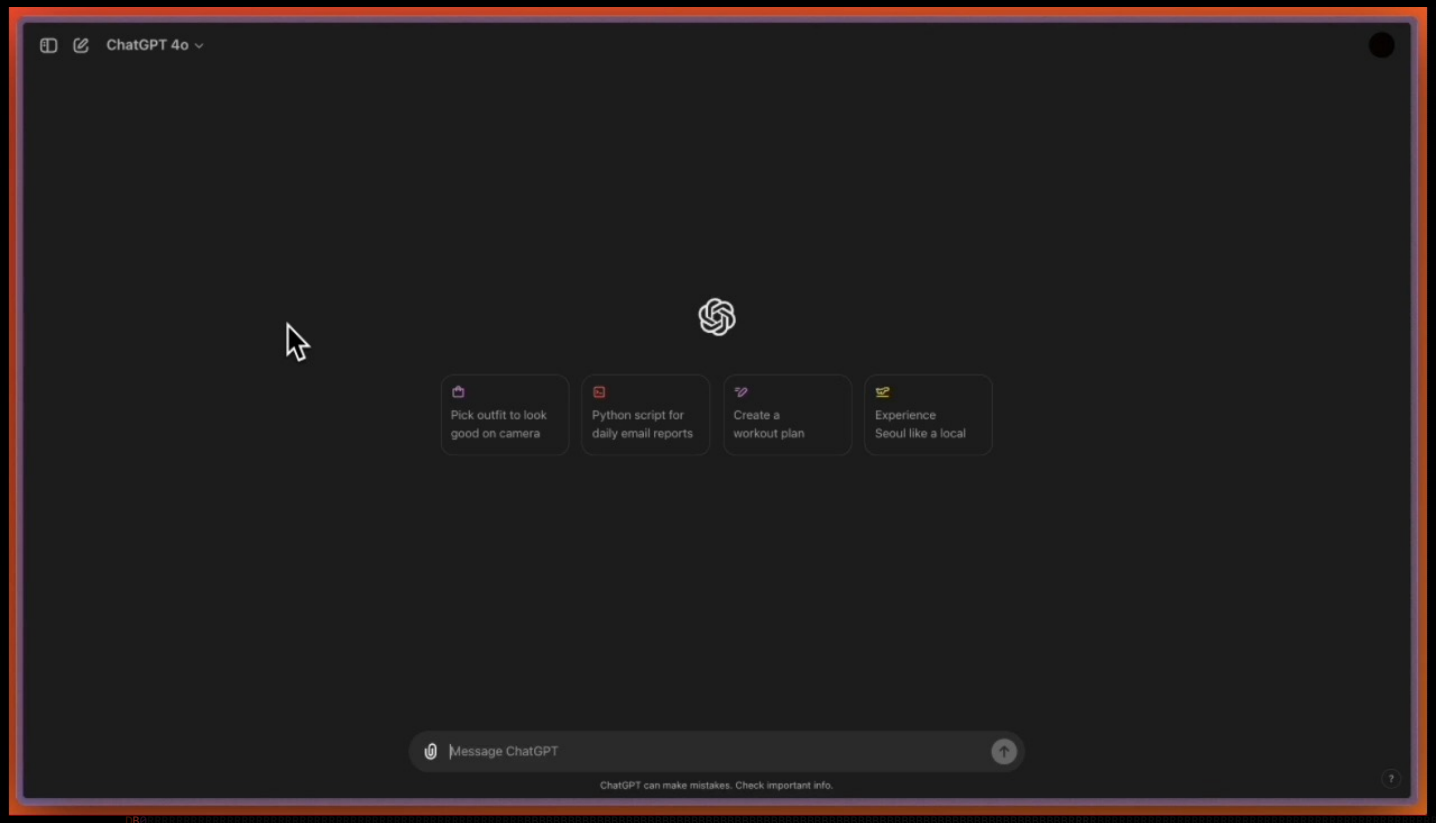
groq



Quality Improvement with Speed.

Multi-agent Improvement

groq


ChatGPT 4o
Pick outfit to look good on camera
Python script for daily email reports
Create a workout plan
Experience Seoul like a local
Message ChatGPT
ChatGPT can make mistakes. Check important info.

groq

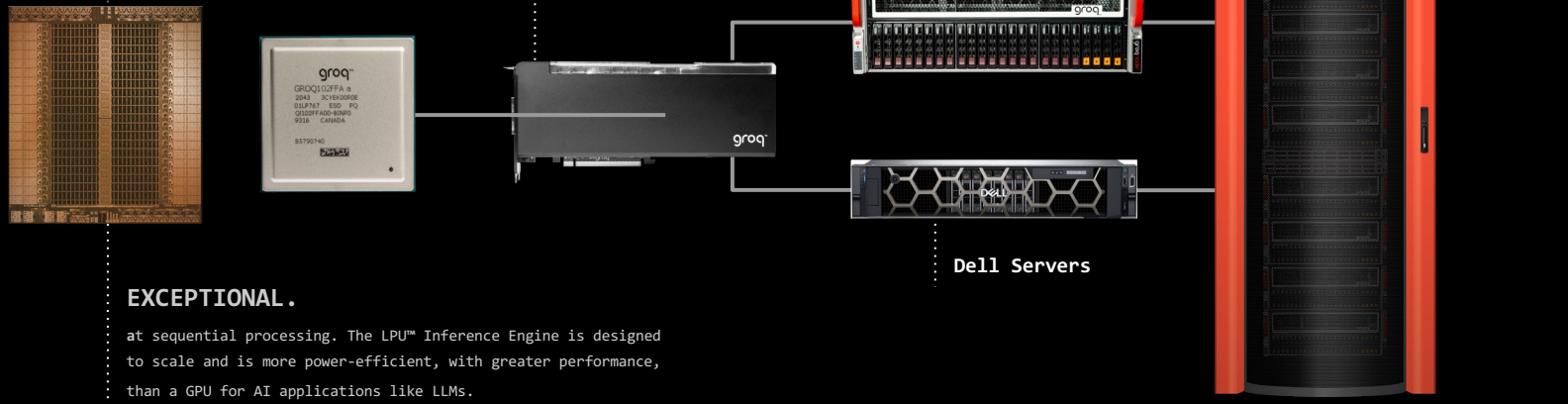
GroqChip™

The purpose-built Language Processing Unit™ accelerator

GroqCard™

GroqNode™

GroqRack™



Dell Servers

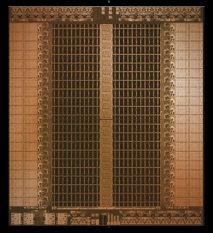
EXCEPTIONAL.

at sequential processing. The LPU™ Inference Engine is designed to scale and is more power-efficient, with greater performance, than a GPU for AI applications like LLMs.

groq

GroqChip™

The purpose-built Language Processing Unit™ accelerator

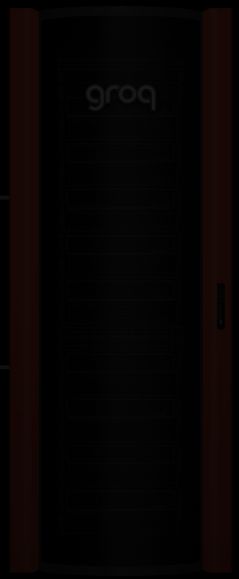


GroqCard™

GroqNode™

Dell Servers

GroqRack™



EXCEPTIONAL.

at sequential processing. The LPU™ Inference Engine is designed to scale and is more power-efficient, with greater performance, than a GPU for AI applications like LLMs.

groq

How Did We Get Here?

End of Dennard Scaling — Single-core ILP hit a wall

Multicore — Shift parallelism responsibility from hardware to software

groq

19% of the instructions are wasted for these benchmarks on an Intel Core i7. The amount of wasted energy is greater, however, since the processor must use additional energy to restore the state when it speculates incorrectly. Measurements like these led many to conclude architects needed a differ-

old observation, called Amdahl's Law, stating that the speedup from a parallel computer is limited by the portion of a computation that is sequential. To appreciate the importance of this observation, consider Figure 5, showing how much faster an application runs with up to 64 cores compared to

Figure 6. Growth of computer performance using integer programs (SPECintCPU).

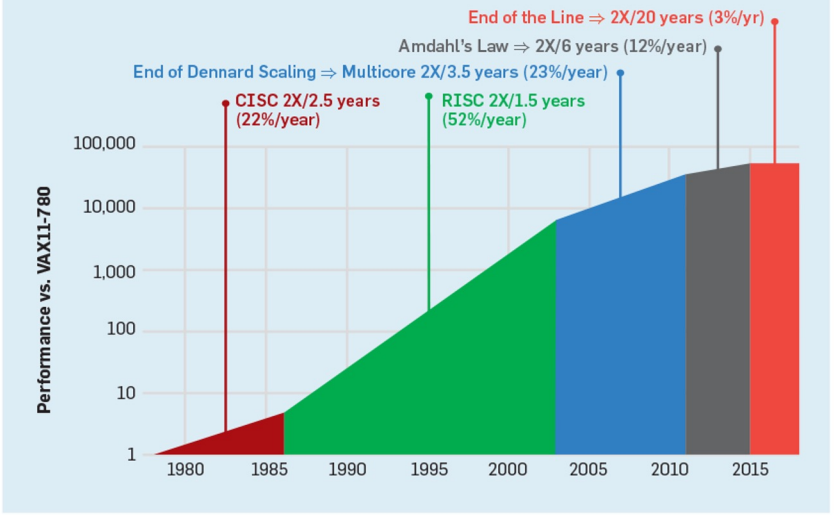
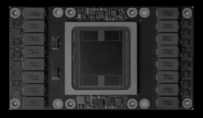


Figure 7. Potential speedup of matrix multiply in Python for four optimizations.

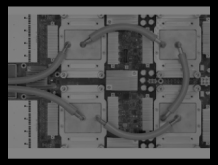
LPU™ Inference Engine

Golden Age of Computer Architecture

Explosion of Domain Specific Architectures (DSA)

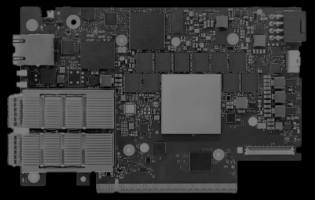


GPU



TPU

VPU



DPU

groq

Algorithms

Dataflow dominated

Statically predictable set of executed operations

Highly-parallel vector operations

Compilers

Remain a challenge

Reliant on hand-tuned libraries

Fragmented front-end ecosystem

Require iterative hardware profiling

Hardware

High-density compute using SIMD

Less silicon area spent on re-ordering and speculation

More memory bandwidth

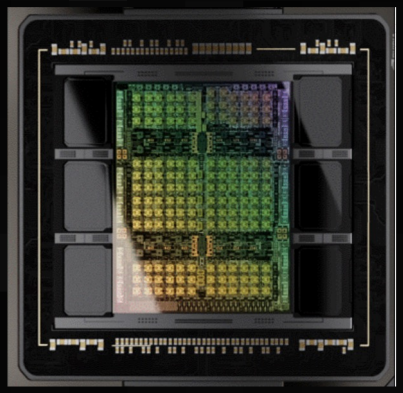
PREDICTABLE

UNPREDICTABLE

Ref. A.Bitar, Presented at Crossroads 3D-FPGA Academic Research Center - December 202

Groq Simplifies Compute

4nm

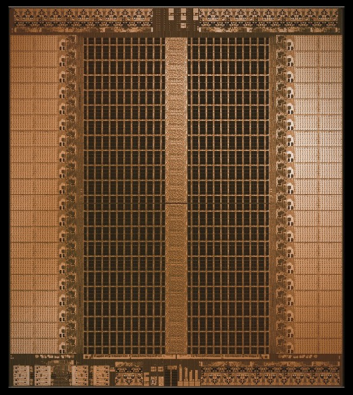


Graphic Processor

Complex

Difficult programming
Less responsiveness
Non-Deterministic execution
Higher costs

14nm



LPU™ Inference Engine

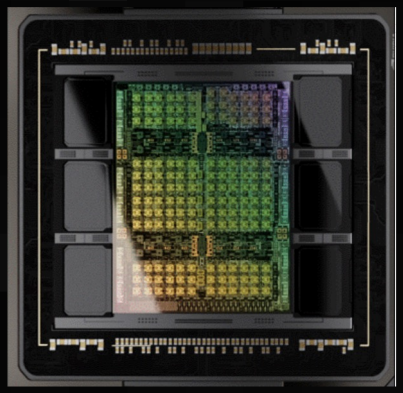
Simplified

Easier compilation
Lower latency
Deterministic / Predictable execution
Massive scale

groq

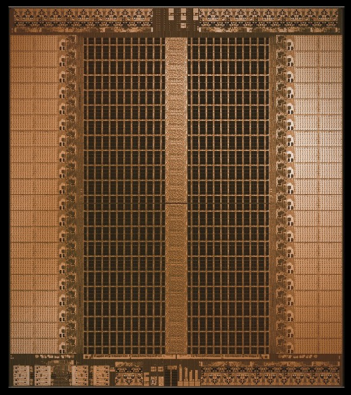
Groq Simplifies Compute

4nm



30X Less Devices

14nm



Graphic Processor

Complex

Difficult programming
Less responsiveness
Non-Deterministic execution
Higher costs

LPU™ Inference Engine

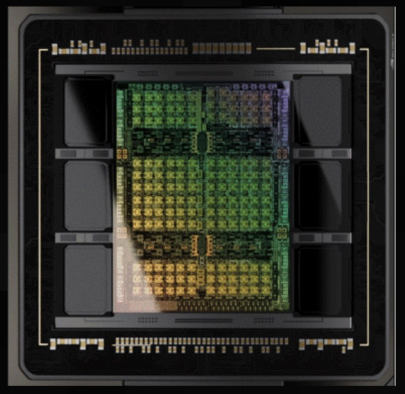
Simplified

Easier compilation
Lower latency
Deterministic / Predictable execution
Massive scale

groq

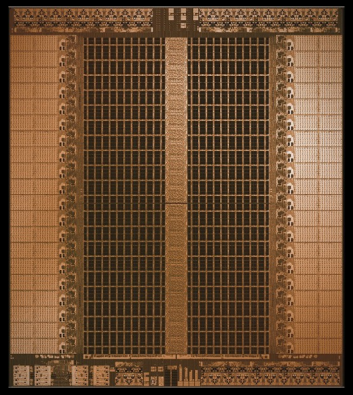
Groq Simplifies Compute

4nm



30X Less Devices

14nm



Graphic Processor

Complex

Difficult programming
Less responsiveness
Non-Deterministic execution
Higher costs

LPU™ Inference Engine

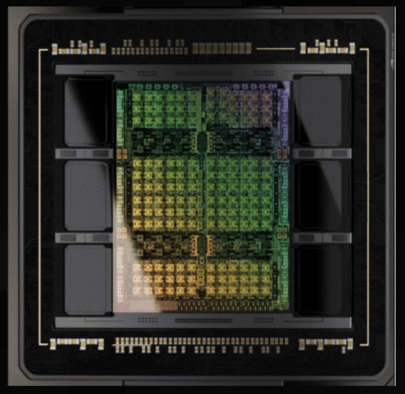
Simplified

Easier compilation
Lower latency
Deterministic / Predictable execution
Massive scale

groq

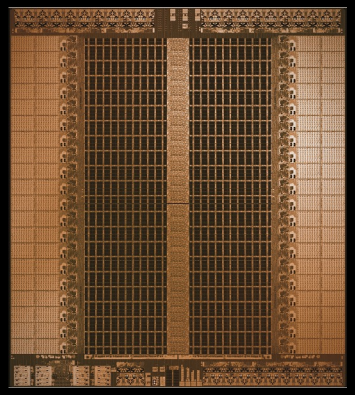
Groq Simplifies Compute

4nm



30X Less Devices

14nm



Graphic Processor

Complex

Difficult programming
Less responsiveness
Non-Deterministic execution
Higher costs

LPU™ Inference Engine

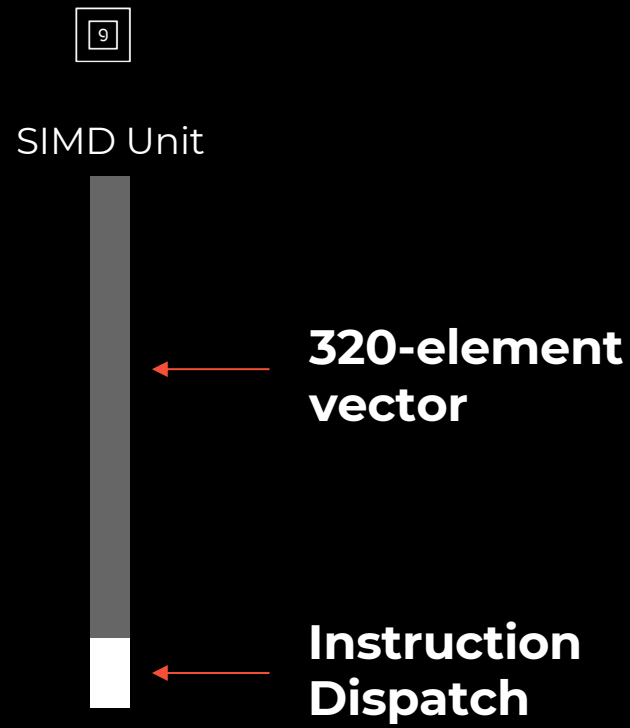
Simplified

Easier compilation
Lower latency
Deterministic / Predictable execution
Massive scale

groq

GroqChip™ Building Blocks

Build different types of specialized SIMD units



GroqChip™ Building Blocks

Build different types of specialized SIMD units

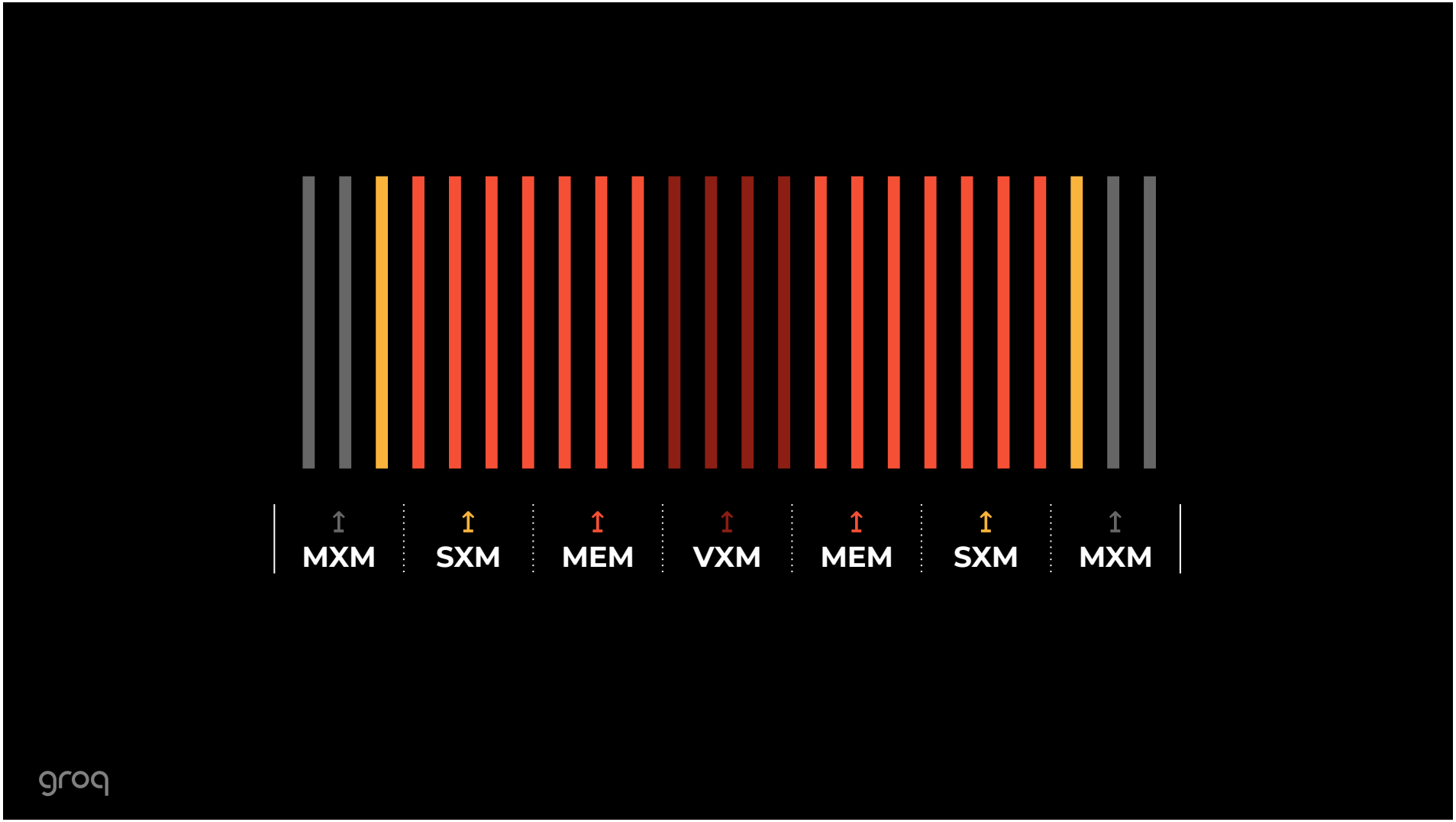
MXM
Matrix-Vector / Matrix-Matrix Multiply

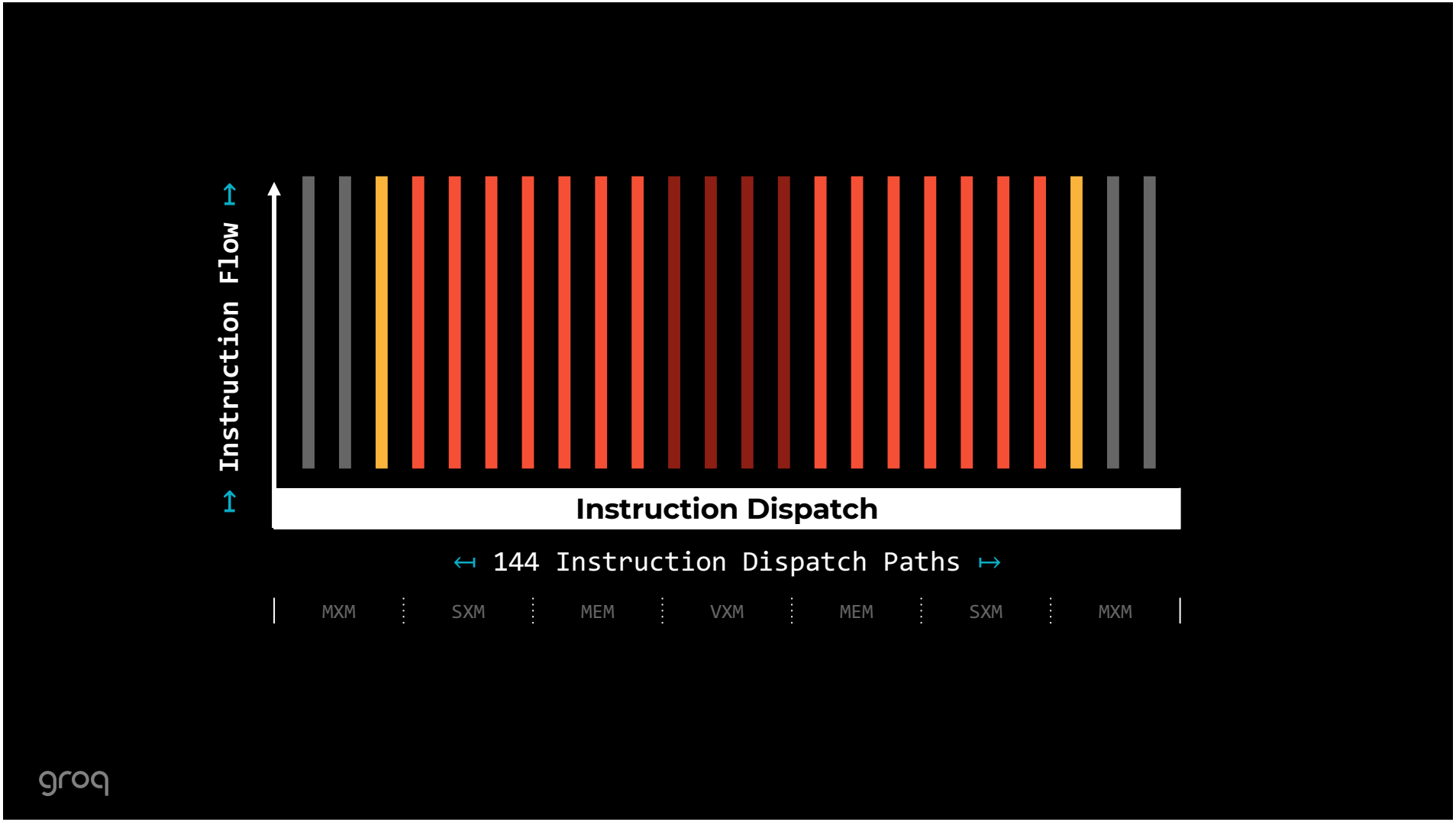
VXM
Vector-Vector Operations

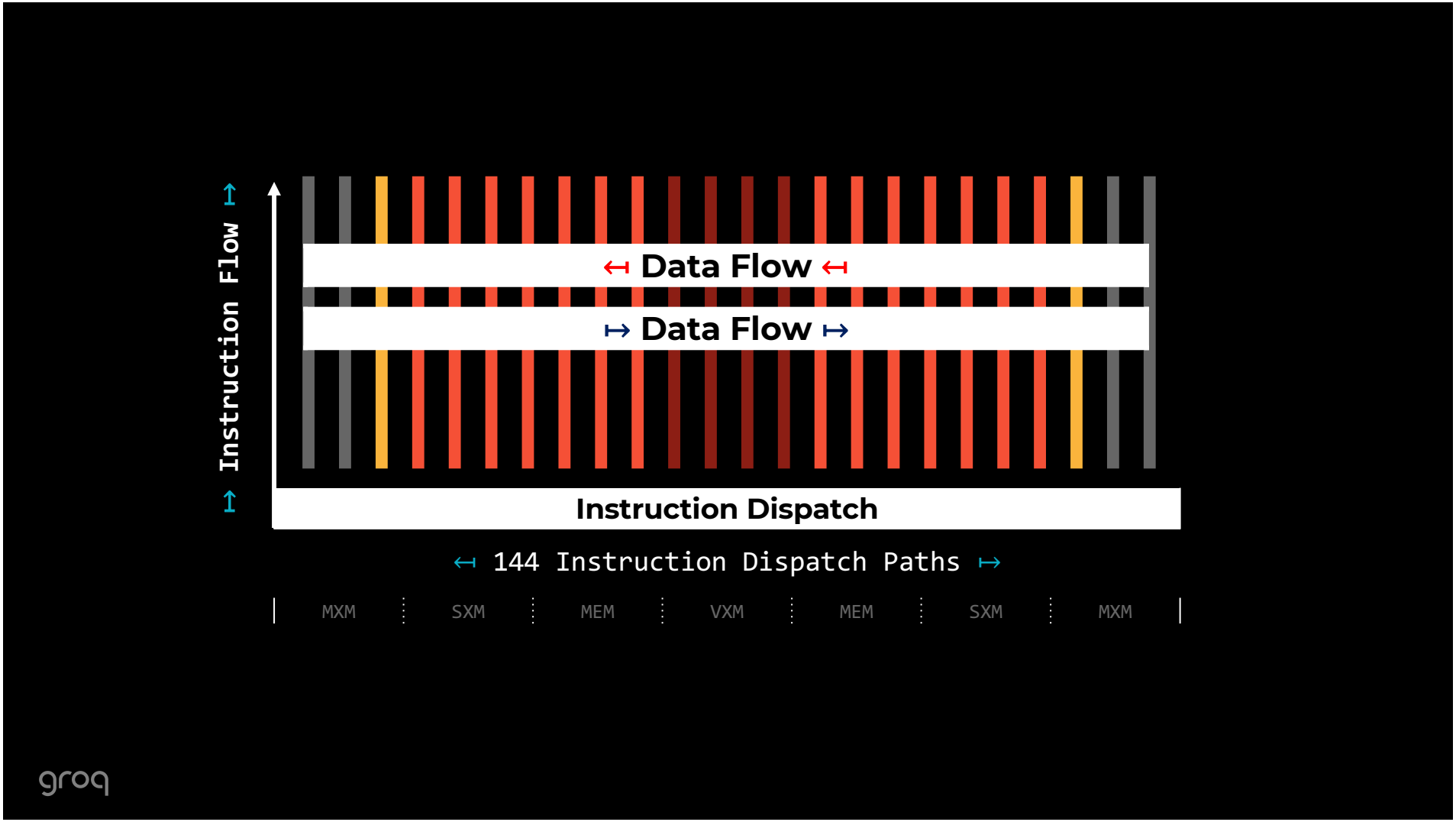
SXM
Data Reshapes

MEM
On-chip SRAM

groq


MXM
SXM
MEM
VXM
MEM
SXM
MXM
groq


Instruction Flow
Instruction Dispatch
144 Instruction Dispatch Paths
MXM
SXM
MEM
VXM
MEM
SXM
MXM
groq


Instruction Flow
Data Flow
Data Flow
Instruction Dispatch
144 Instruction Dispatch Paths
MXM
SXM
MEM
VXM
MEM
SXM
MXM
groq

Empowering

Groq™ Compiler

groq

Architecture Empowering Software

Software-controlled memory

No dynamic hardware caching

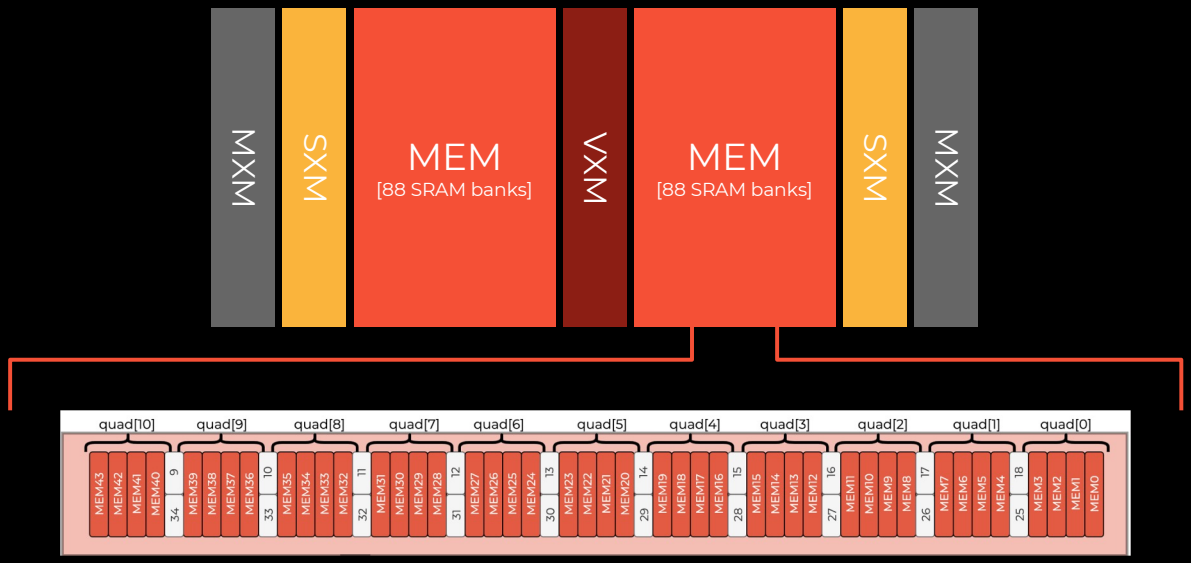
- Compiler aware of all data locations at any given point in time

Flat memory hierarchy
(no L1, L2, L3, etc)

- Memory exposed to software as a set of physical banks that are directly addressed

Large on-chip memory capacity (220 MiB) at very high-bandwidth (80 TBps)

- Achieves high compute efficiency even at low operational intensity



groq

Architecture Empowering Software

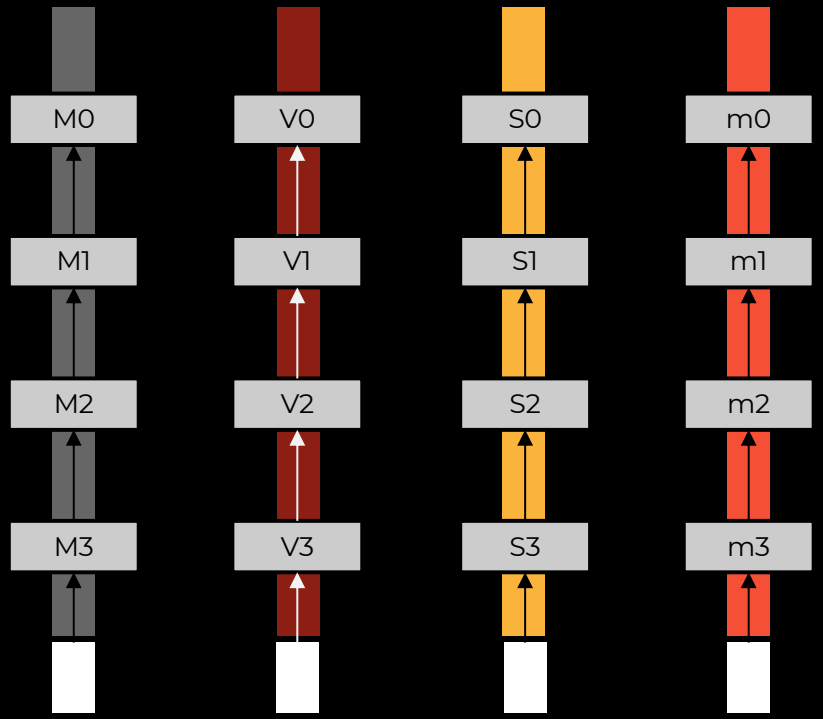
Lockstep execution of Functional Units

Compiler empowered to perform cycle-accurate instruction scheduling

- Synchronous “threads”
- One instruction issued per cycle at each dispatch path

Little hardware control needed for managing instruction execution

- < 3% area overhead for instruction dispatch logic



groq

Architecture Empowering Software

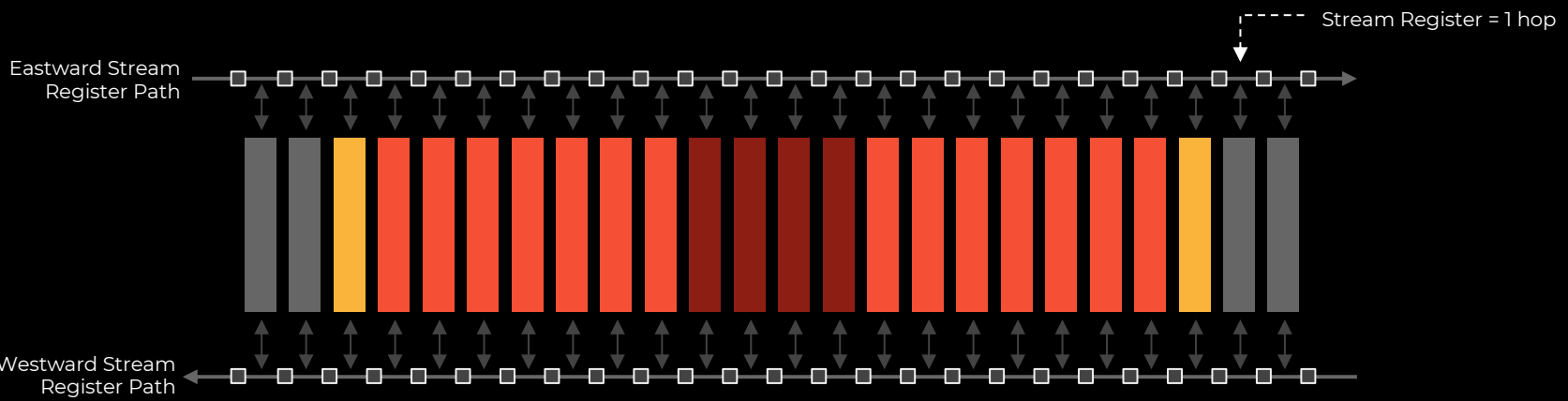
Simple, one-dimensional interconnect for inter-FU communication

Compiler can quickly reason about all data movement between FUs

- Eastward and westward paths made up of arrays of “stream registers”
- Stream register = one-cycle hop

No arbiters / queues = software can easily reason about exact data movement without simulation

Travel time calculation as simple as a single add/subtract



groq

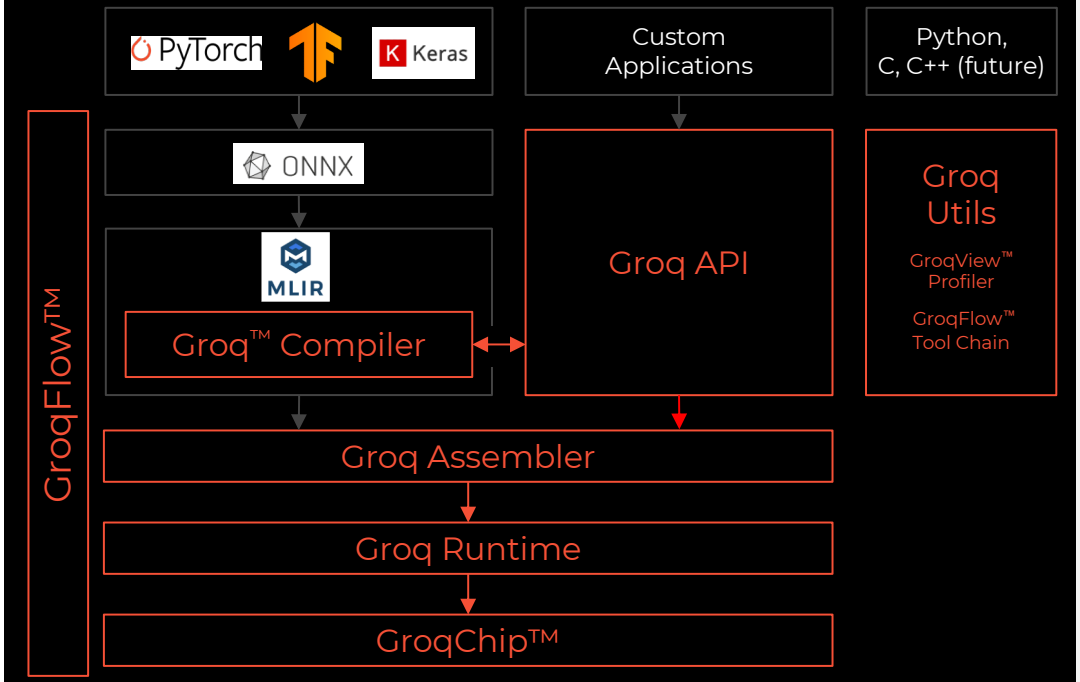
Instruction Set

Low level,
320 element vector ops,
explicit resource selection

Function	Instruction	Description
ICU	`NOP N`	No-operation, can be repeated `N` times to delay by `N` cycles
	`Ifetch`	Fetch instructions from streams or local memory
	`Sync`	Parks at the head of the instruction dispatch queue to await barrier notification
	`Notify`	Releases the pending barrier operations causing instruction flow to resume
	`Config`	Configure low-power mode
	`Repeat n,d`	Repeat the previous instruction `n` times, with `d` cycles between iterations
MEM	`Read a,s`	Load vector at address `a` onto stream `s`
	`Write a,s`	Store stream `s` register contents into main memory address `a`
	`Gather s, map`	Indirectly read addresses pointed to by `map` putting onto stream `s`
	`Scatter s, map`	Indirectly store stream `s` into address in the `map` stream
	`Countdown d`	Set the delay `d` in cycles between loop iterations
	`Step a`	Set the stride `a` between subsequent generated memory addresses
	`Iterations n`	Set the loop bounds for address generation
VXM	unary operation	`z=op x` point-wise operation on 1 operand, `x`, producing 1 result, `z` (eg. `mask, negate`)
	binary operation	`z=x op y` point-wise operations with 2 operands *x* and *y* producing 1 result, (e.g. add, mul, sub)
	type conversions	Converting fixed point to floating point, and vice versa
	`ReLU`	Rectified linear unit activation function max(0,x)
	`TanH`	Hyperbolic tangent - activation function
	`Exp`	exponentiation e^{π}
	`RSqrt`	Reciprocal square root
MXM	`LW`	Load weights (LW) from streams to weight buffer
	`IW`	Install weights (IW) from streams or LW buffer into the 320 x 320 array
	`ABC`	Activation buffer control (ABC) to initiate and coordinate arriving activations
	`ACC`	Accumulate (ACC) either INT32 or FP32 result from MXM
SXM	`Shift up/down N`	Lane-shift streams up/down by *N* lanes
	`Permute map`	Bijective permute 320 inputs ➔ outputs
	`Distribute map`	Rearrange or replicate data within a superlane (16 lanes)
	`Rotate stream`	Rotate `n x n` input data to generate n^2 output streams with all possible rotations (`n`=3 or `n`=4)
	`Transpose sg16`	Transpose 16x16 elements producing 16 output streams with rows and columns interchanged
C2C	`Deskew`	Manage skew across plesiochronous links
	`Send`	Send a 320-byte vector
	`Receive`	Receive a 320-byte vector, emplacing it in main memory

groq

GroqWare™ Suite



groq

DIVERSE SUITE OF DEVELOPMENT TOOLS

Out-of-Box

Groq Compiler provides out-of-box support for standard Deep Learning models

Fine Grained Control

Groq API provides finer grained control of GroqChip in order to support custom applications

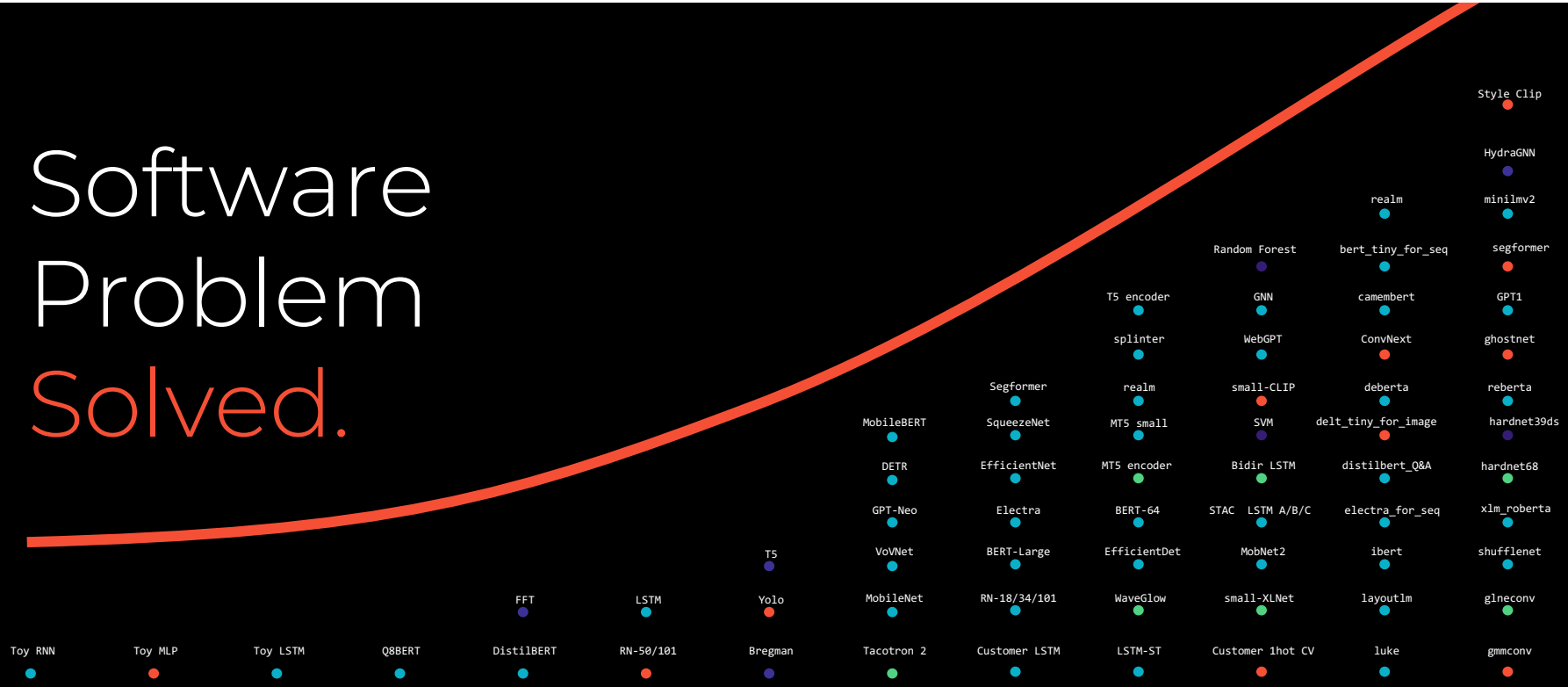
+

Productivity Tools

GroqView Profiler provides visualization of the chip's compute and memory usage at compile time

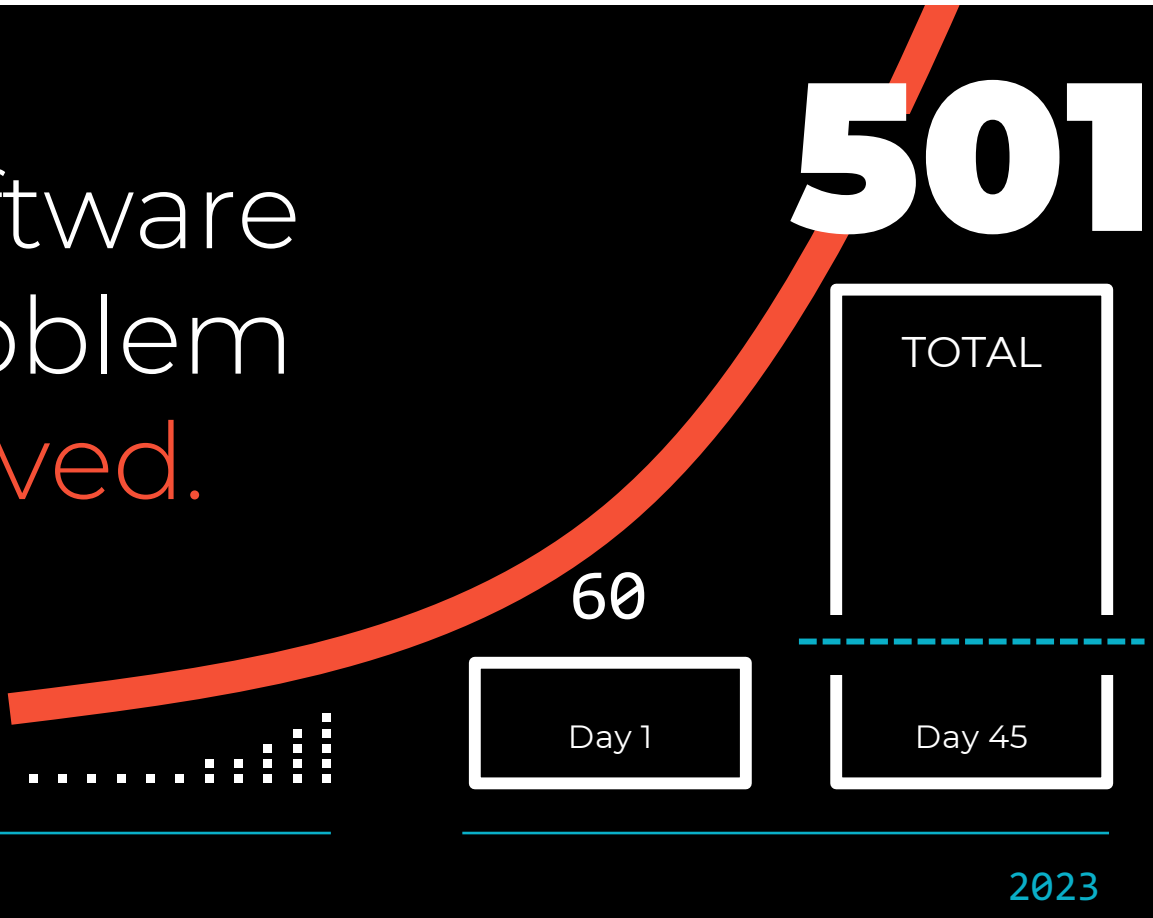
GroqFlow Tool Chain enables a single line of Pytorch or TensorFlow code to import and transform models through a fully automated tool chain to run on Groq hardware

Software Problem Solved.



groq

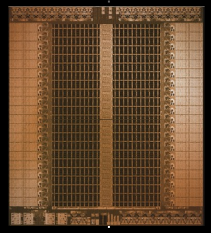
Software
Problem
Solved.




Llama3 70B
In production
10 hours
The same day
groq®

GroqChip™

The purpose-built Language Processing Unit™ accelerator

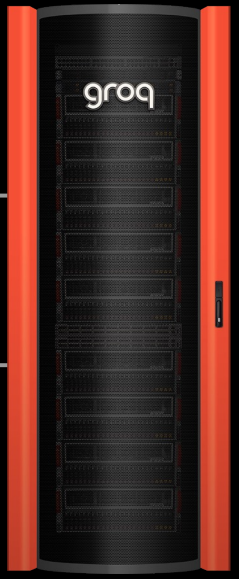


GroqCard™

GroqNode™

Dell Servers

GroqRack™



EXCEPTIONAL.

at sequential processing. The LPU™ Inference Engine is designed to scale and is more power-efficient, with greater performance, than a GPU for AI applications like LLMs.

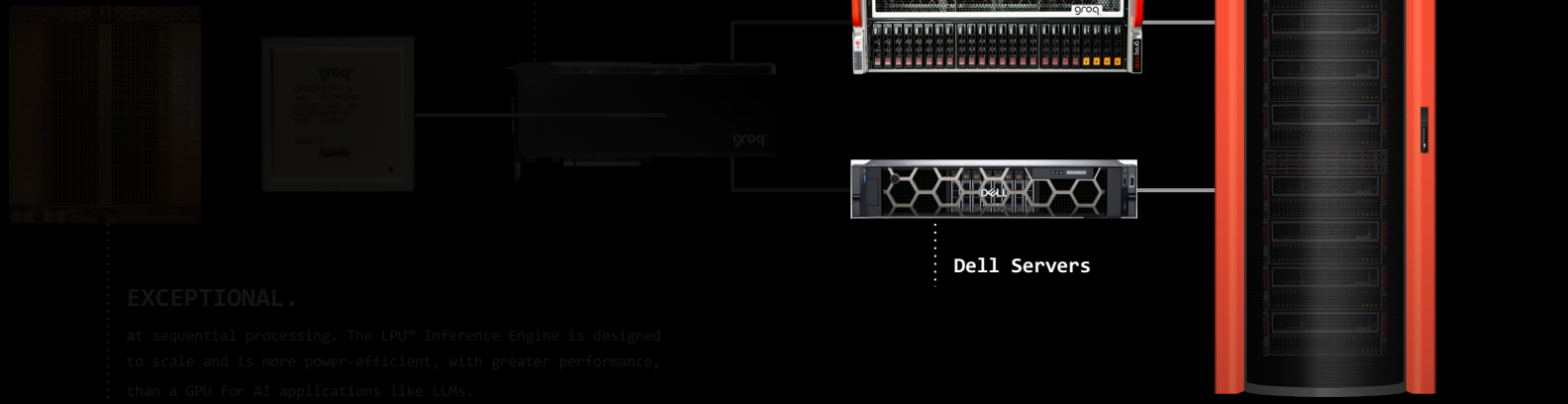
GroqChip™

The purpose-built Language Processing Unit™ accelerator

GroqCard™

GroqNode™

GroqRack™



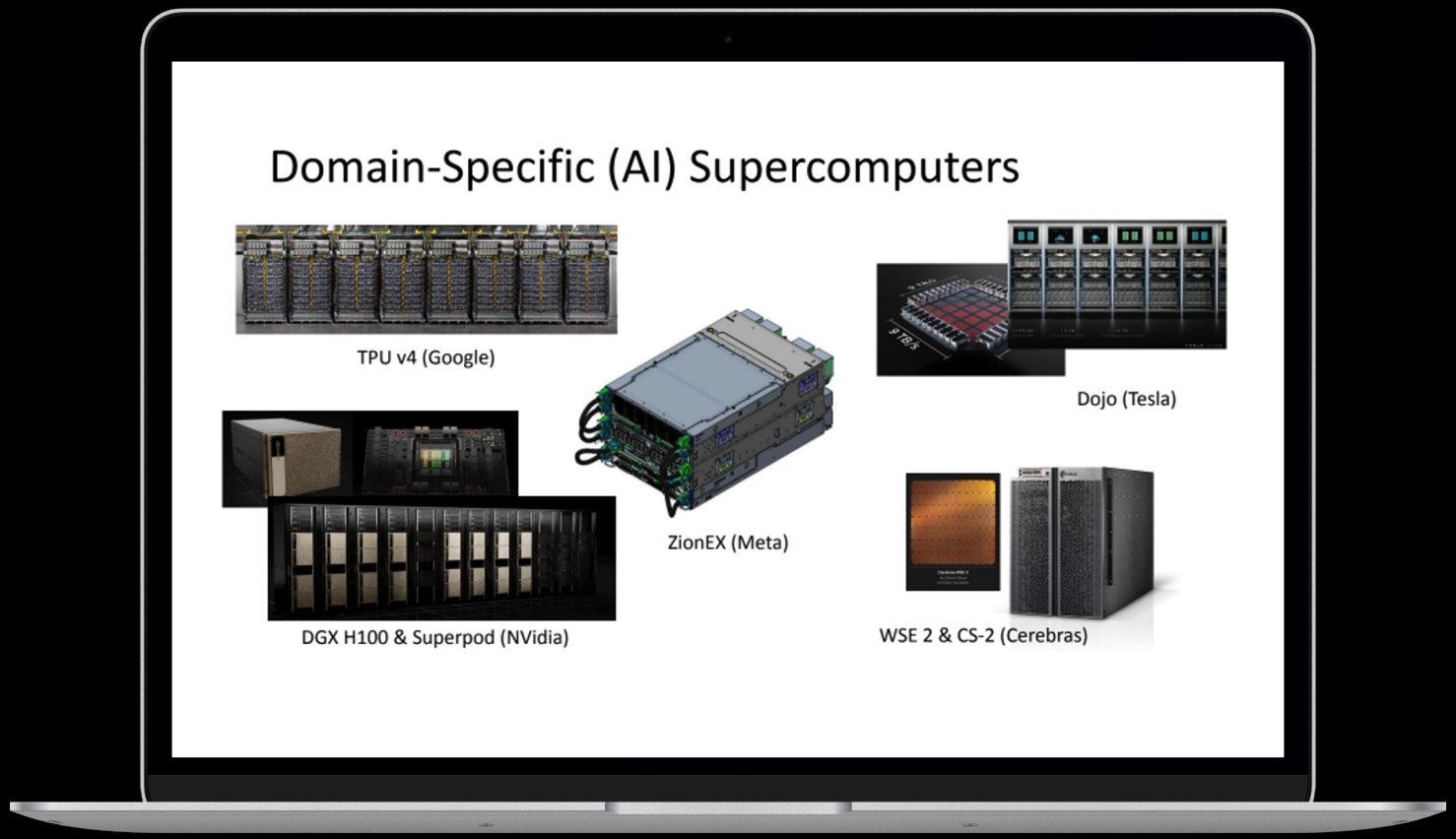
Dell Servers

EXCEPTIONAL.

at sequential processing. The LPU™ Inference Engine is designed to scale and is more power-efficient, with greater performance, than a GPU for AI applications like LLMs.

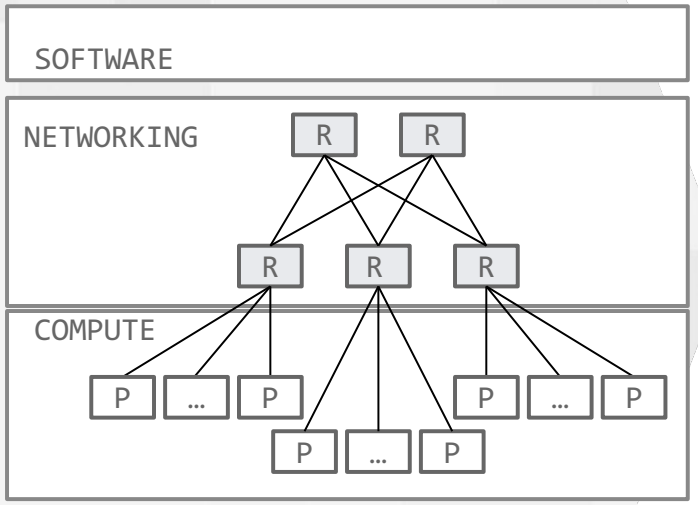
groq

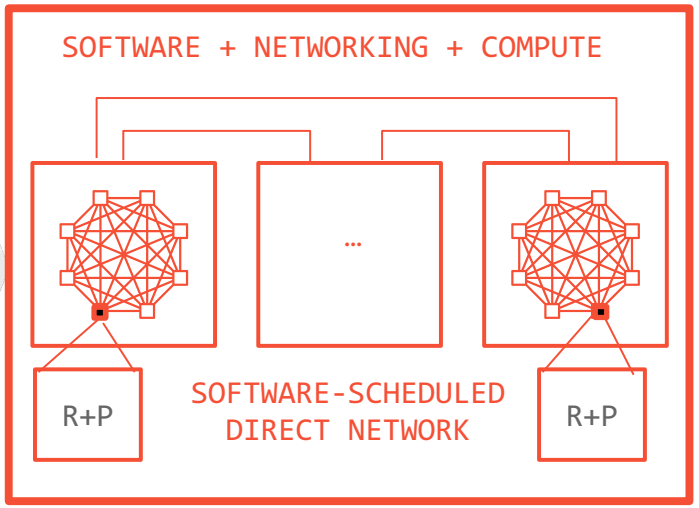
The explosion of
Domain-Specific Networks



groq

Groq Simplifies Interconnect





Conventional Network

Disjoint Compute/Networking

COMPLEX

Per-hop Router arbitration

(High Latency)

Hardware-based global adaptive routing

(Weak Scaling)

Congestion sensing in the network through backpressure

(Non-deterministic delay)

High network load sensitivity

SW Controlled Network

SW Orchestrated Compute & Network

SIMPLIFIED

No hardware-arbitration for dynamic contention

(Lower latency)

No hardware routing

(Massive Scale)

No congestion sensing

(Deterministic Delay)

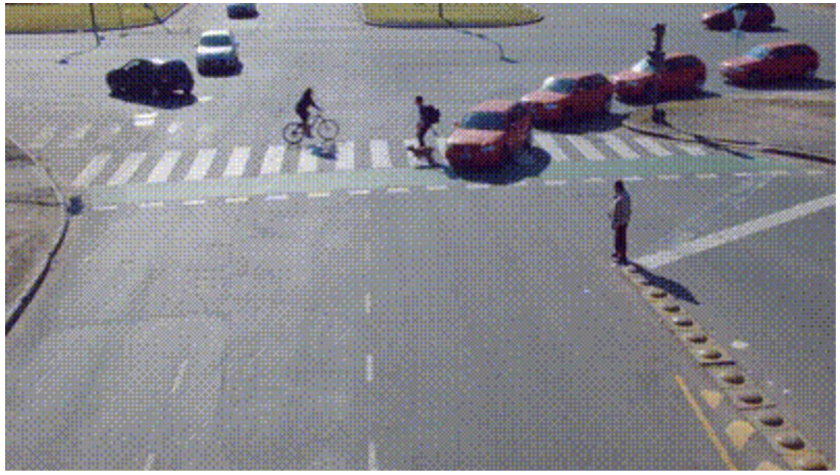
Low network load sensitivity

groq

Traffic Nightmare

Stop-and-go bumper-to-bumper traffic gives **long travel times** and **poor gas mileage**

The unpredictable nature leads to **poor utilization** of roads



A Smart City for AI

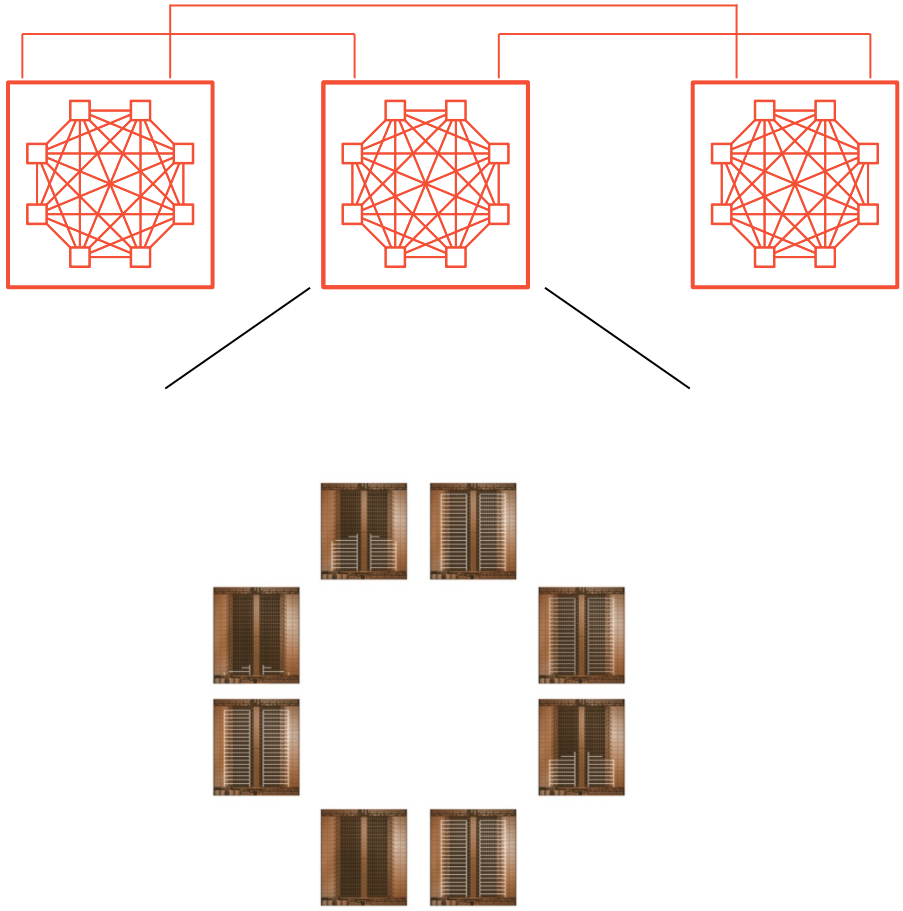
Non-stop point-to-point transport gives the **shortest travel time** and **best gas mileage**

Predictability means **high utilization** and no accidents **EVER**

Nothing is left to chance

Groq's Software-scheduled Network.

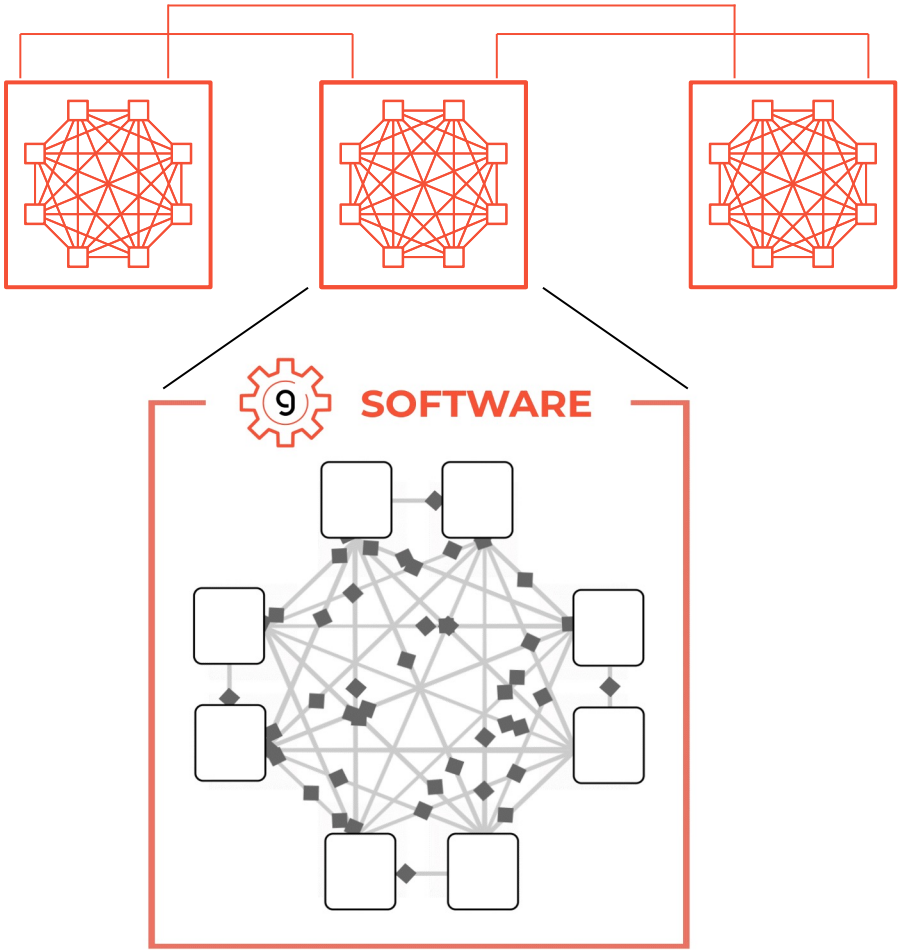
- RealScale™ chip-to-chip (C2C) interconnect enables synchronous communication
- Groq network synchronizes all chips to work on the same global clock acting like a single core cluster
- Clock drift across chips is accounted for and mitigated deterministically



groq

Groq's Software-scheduled Network.

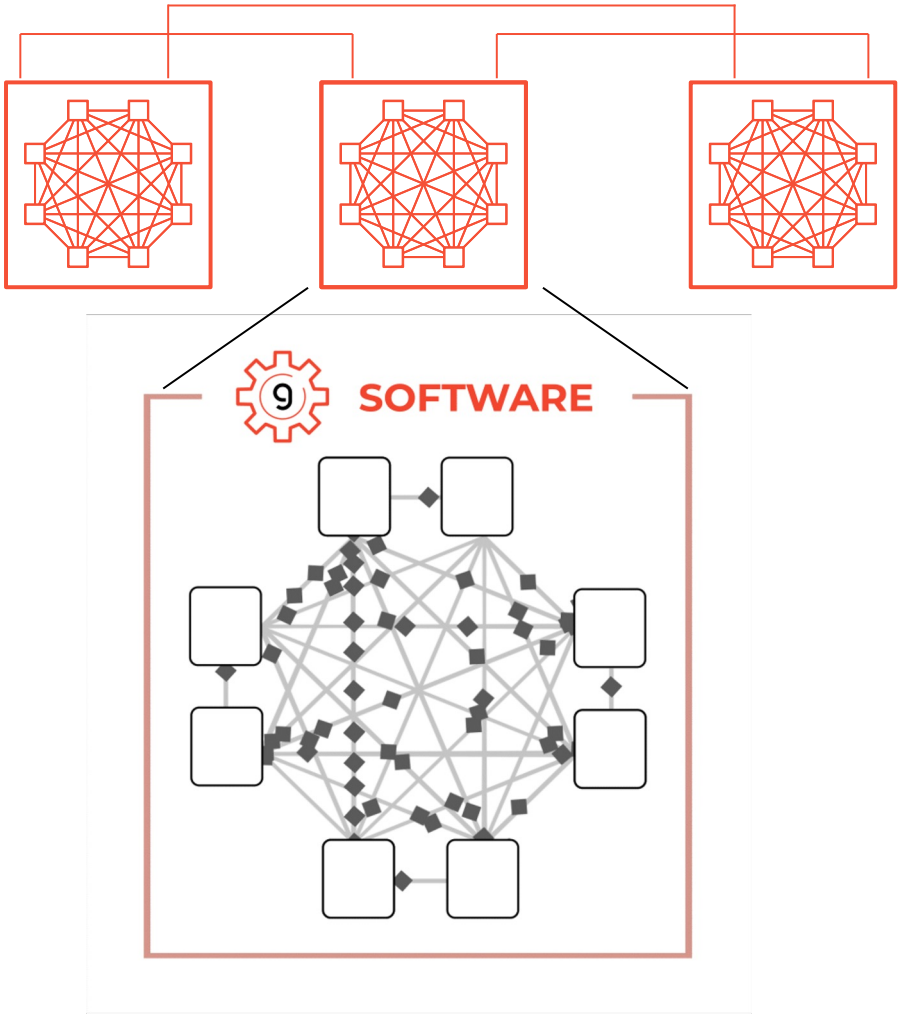
- Each chip acts as both a processor and router
- Compiler schedules messages as part of programs loaded onto each chip
- No adaptive routing / congestion sensing needed



groq

Groq's Software-scheduled Network.

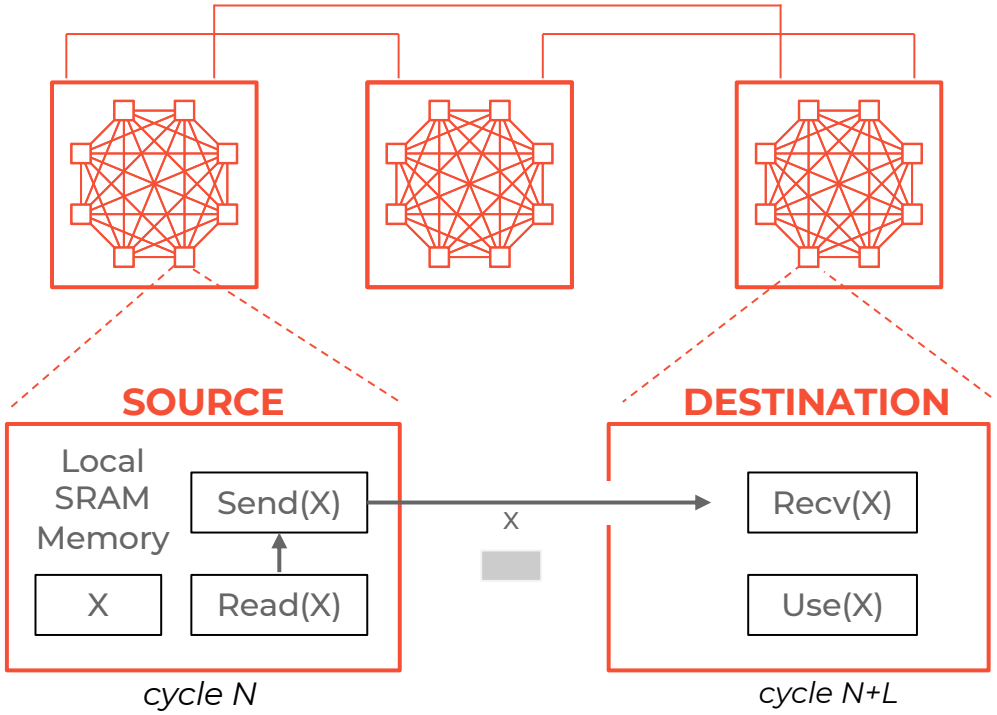
+ Software controlled networking improves efficiency to connect two nodes through both minimal and non-minimal paths



groq

Groq Simplifies Compute.

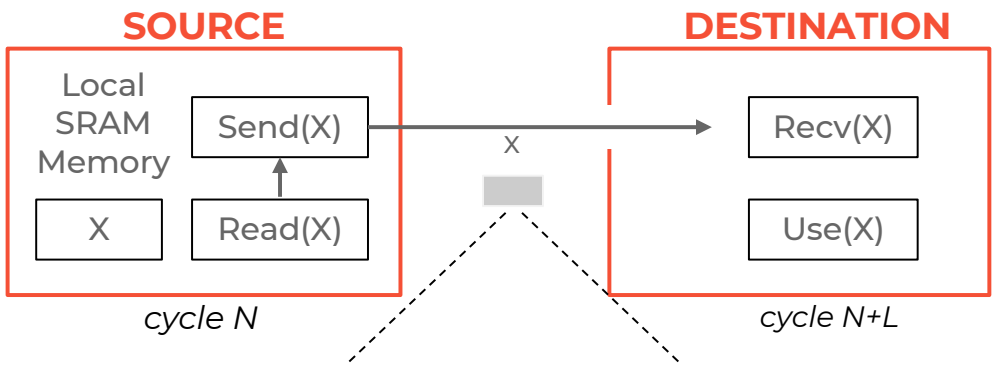
- Compiler knows exact cycle data should be sent from one chip and received at another
- Each GroqChip has access to 2TB of global memory (across 10K chips) accessible in < 3us end-to-end system latency

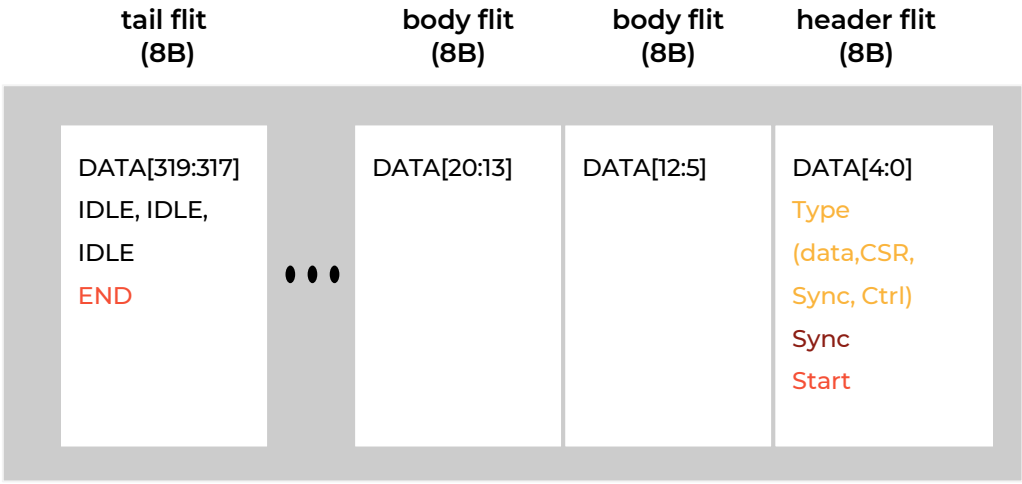


groq

Routing Tensors, Not Packets.

- Packet format: 320-byte vector
- Only 2.5% encoding overhead, from the header+tail flit
- No hardware flow control, no virtual channel information, etc.
- Improves performance (bandwidth) especially at smaller message (tensor) size





groq

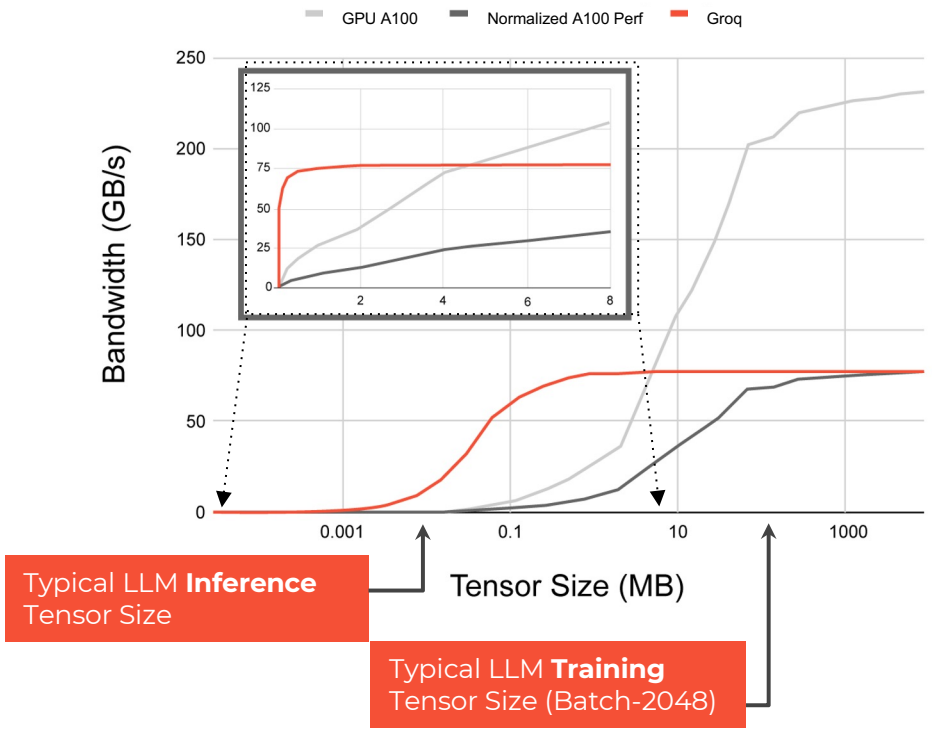
AllReduce Comparison Results.

Comparison made with an eight GPU A100 system with NCCL

+ Only a handful of cycles to Read(vector) → Send(vector) enables fine-grained communication across the 16 directly connected links on each TSP

+ A100 system has approximately 3x higher network channel bandwidth

+ When normalized, Groq TSP matches the bandwidth at large tensor size while significantly improving bandwidth at intermediate tensor size

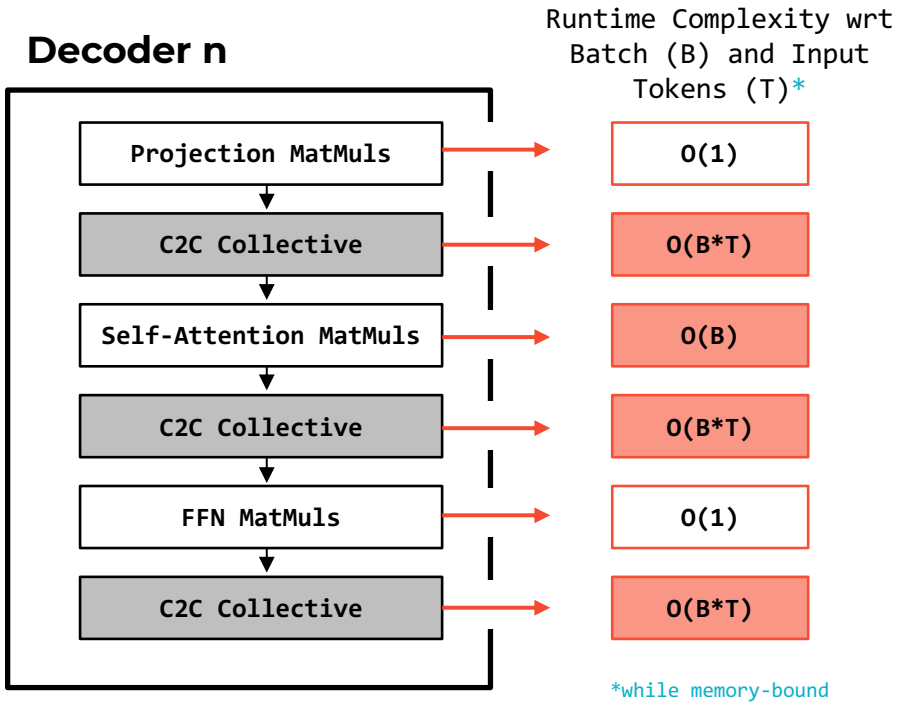


groq

Results for A100 were measured on an 8 A100 GPU system with 300 GB/s of NVlink bandwidth per GPU connected through NVSwitch. Nvidia results are from publicly available data on https://github.com/NVIDIA/nccl-tests/

Transformer Decoder Architecture

- Transformer-based LLM inference is memory-bound
- BLAS-2 (vector-matrix multiplies)
- LLMs are often too large to fit in a single device's memory
- Requires collectives (e.g. all-reduce)
- Batching improves the efficiency of Projection and FFN
- Scaling cost of self-attention and collectives



groq

Results for A100 were measured on an 8 A100 GPU system with 300 GB/s of Nvlink bandwidth per GPU connected through NVSwitch. Nvidia results are from publicly available data on https://github.com/NVIDIA/nccl-tests/

Scaling Transformer Inference

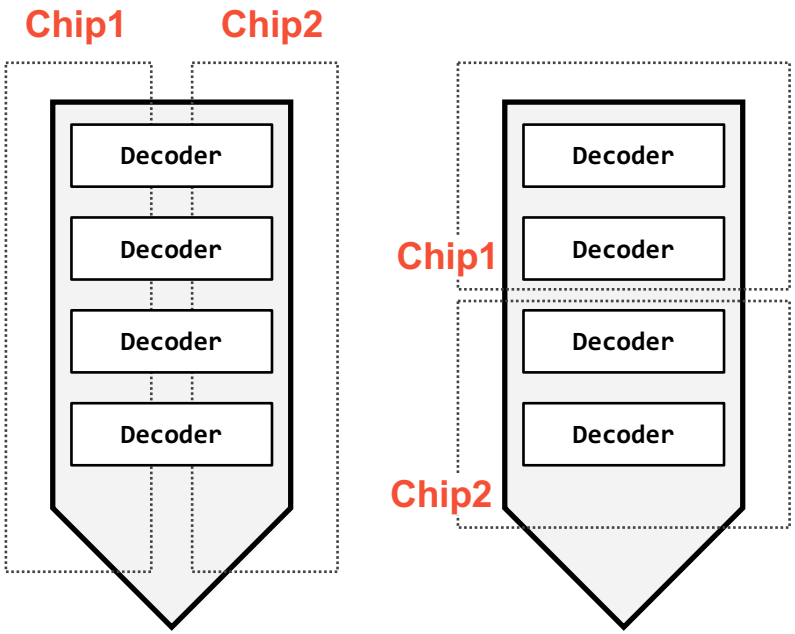
Tensor / Model Parallelism (TP)

Benefits

- Reduces latency per token
- Increases memory capacity

Drawbacks

- Introduces C2C collectives
- Harder to shard evenly compared to PP



Pipeline Parallelism (PP)

Benefits

- Increases memory capacity
- Small C2C cost for pipeline transfers compared to TP collectives
- Easy to shard

Drawbacks

- No latency improvement

groq

Throughput / $

$$\text{tok/s/\$} = \frac{T}{L \times C \times \$}$$

T = tokens concurrently processed by system

L = latency of system

C = chips per system

$ = Cost per Deployed Chip

LPU

Latency (L)

High-bandwidth SRAM reduces L ↦ lowering the required T/C to saturate compute

Tokens per chip (T / C)

Fill pipeline with active users (Batching and Pipelining)

Low Batching ↦ cheaper self-attention& collectives ↦ lower L (improving tok/s/$)

Low LPU Cost per Chip Deployed

CAPEX: Low-cost implementation in 14nm, built-in switching (no external NICs/ NVSwtiches, etc)

OPEC: SRAM power/access 20x lower than HBM

GPU

Latency (L)

Low bandwidth HBM (1/100 of SRAM) increases L ↦ requires higher T/C to saturate compute ↦ More Batching

Tokens per chip (T / C)

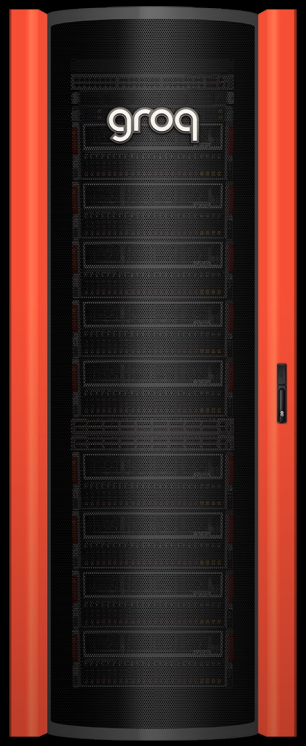
More Batching ↦ More expensive self-attention & collectives ↦ even higher L (diminishing returns)

High GPU Cost per Chip Deployed

CAPEX: Expencive state of art Module (HBM, CoWoS, Reticle Sized Compute Die) with 30x more devices + Additional NICs / NVSwitches, etc

OPEX: HBM power/access 20x higher than SRAM

groq



Groq Enables Lowest-Latency LLM inference :

- Chip Determinism unlocks performance and efficiency while paving the way for more compute density with 3D chip integration
- Low-diameter dragonfly network with abundant path diversity achieves better latency and peak AllReduce performance
- Synchronous global communication extends multiple GroqChips into a massive multi-chip single-core cluster
- Software Scheduled Deterministic Compute and Network uses Global time to provide efficient C2C utilization enabling massive scale

groq®

What's Next.


groq®