Cerebras Systems

The Power of Wafer-Scale Systems

cerebras

Cerebras Systems

Building a new class of computer system for the future of AI and HPC

A full AI acceleration solution: chip, system, software, ML

Founded in 2016

350+ Engineers

Offices
Silicon Valley | San Diego | Toronto | Bangalore

Customers
North America | Asia | Europe

Our Partners and Customers

Large Enterprise, HPC, Government

- G42, Mayo Clinic, GlaxoSmithKline, TotalEnergies, AstraZeneca, Bayer, Genentech, Tokyo Electron Devices...
- ANL, LLNL, NETL, PSC, NCSA, EPCC, Leibniz Supercomputing Centre...
- Security, e.g. DARPA, USAF, ARL

GSK

AstraZeneca

MAYO CLINIC

PSC

Genentech
A Member of the Roche Group

TotalEnergies

Argonne NATIONAL LABORATORY

Opentensor

G42

BAYER

TOKYO ELECTRON

NATIONAL ENERGY TECHNOLOGY LABORATORY

DARPA

Leibniz Supercomputing Centre
of the Bavarian Academy of Sciences and Humanities

Lawrence Livermore National Laboratory

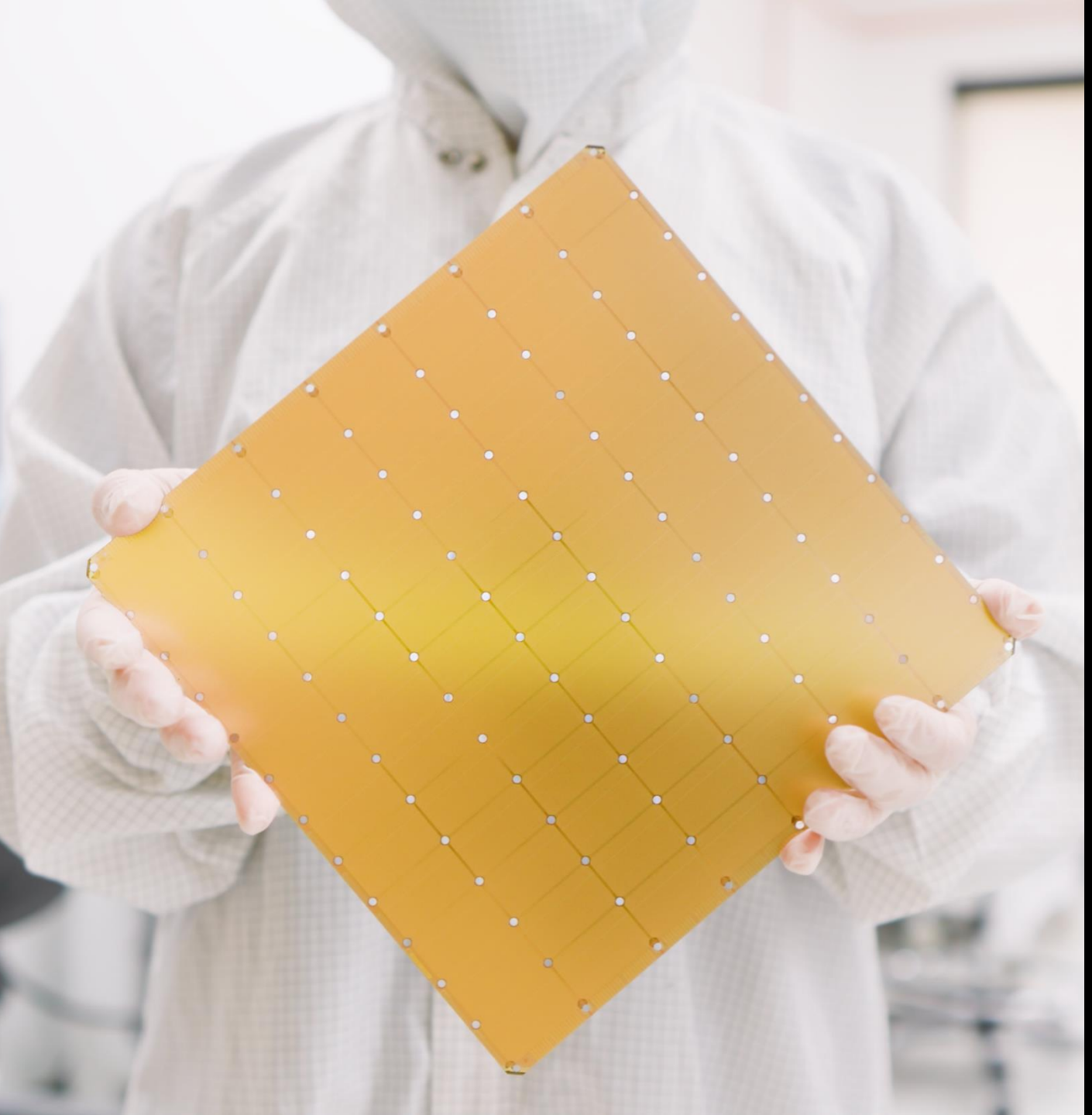
National Center for Supercomputing Applications
UNIVERSITY OF ILLINOIS URBANA-CHAMPAIGN

epcc

ARL

UNITED STATES AIR FORCE

JOINT SPECIAL OPERATIONS COMMAND



Cerebras Wafer-Scale Engine 3

The fastest AI chip on earth again

125 PFLOP/s of sparse AI compute

46,225 mm^2 silicon

4 trillion transistors

5nm TSMC process

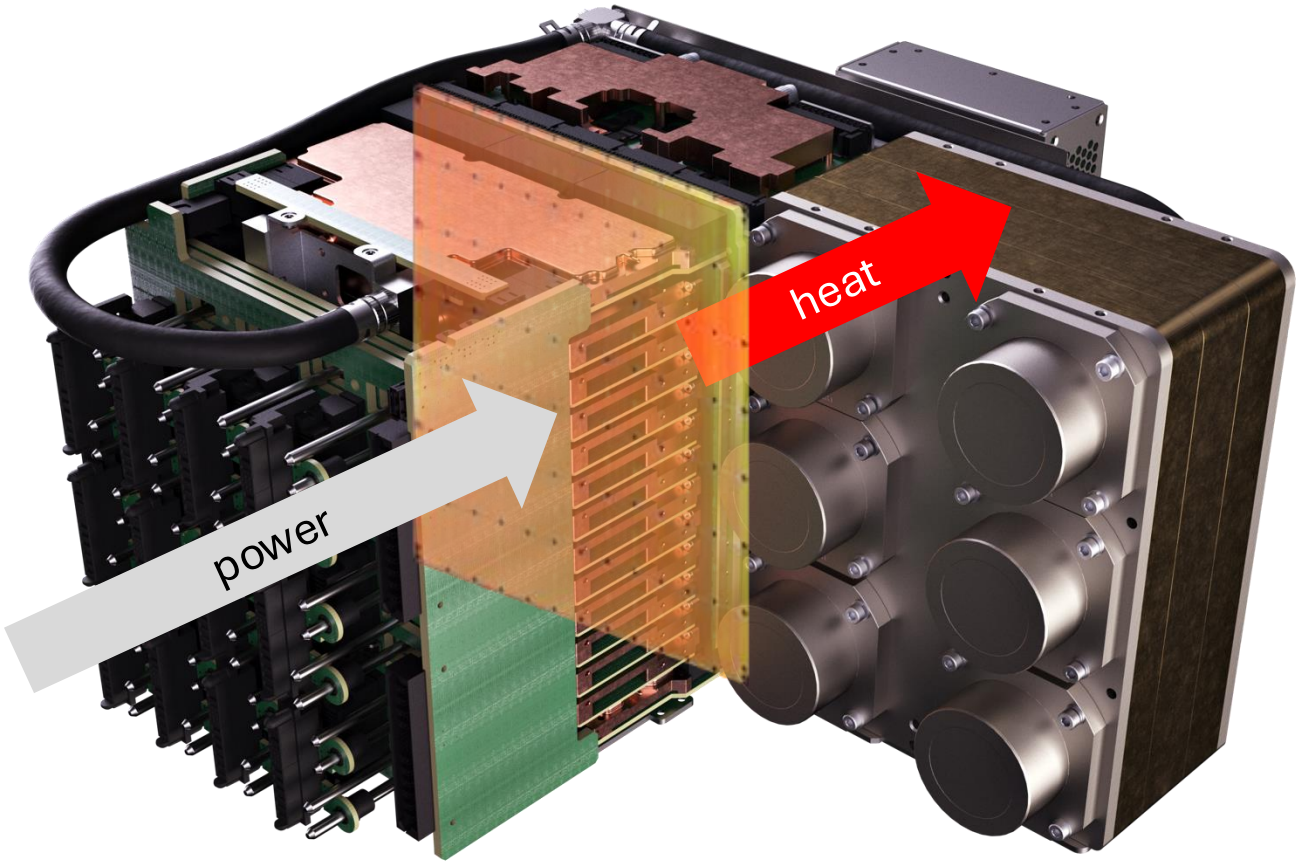
900,000 cores optimized for sparse linear algebra

45 GB of on-chip memory

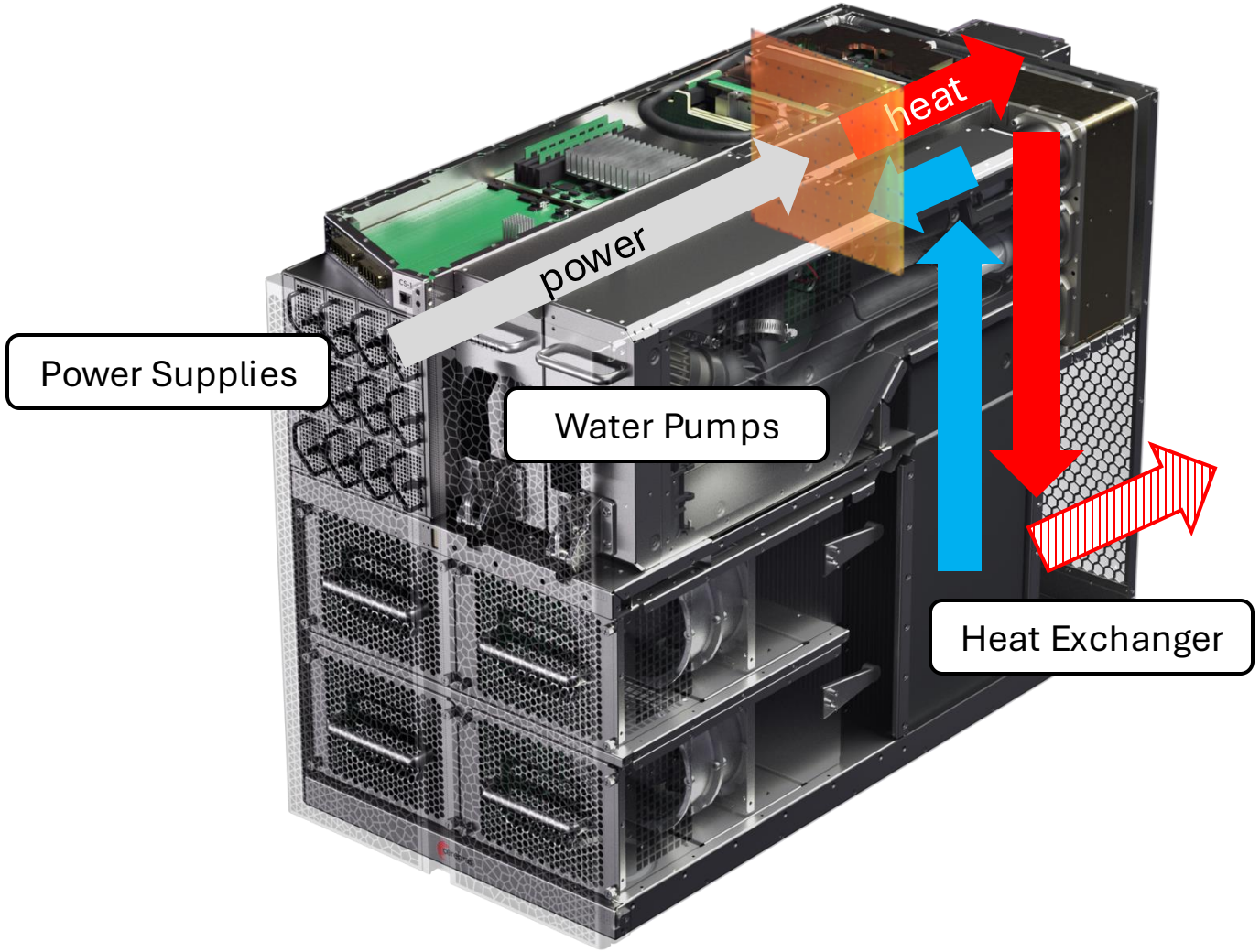
25 PB/s memory bandwidth

30 PB/s fabric bandwidth

The Cerebras System: Engine Block



The Cerebras System: Chassis



cerebras

cerebras
cerebras

cerebras

Santa Clara, California

8 Exaflops

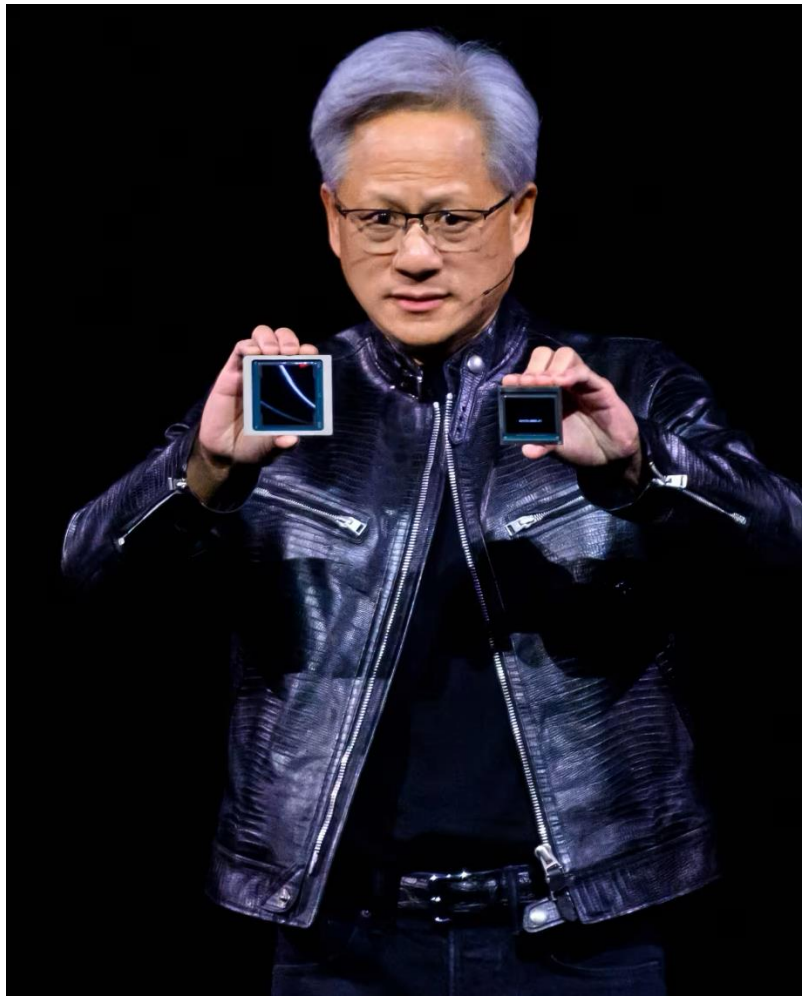
Dallas, Texas

20 Exaflops

Minneapolis, MN
64 ExaFLOPs

Inference Deep Dive

Bigger chips are faster and more efficient

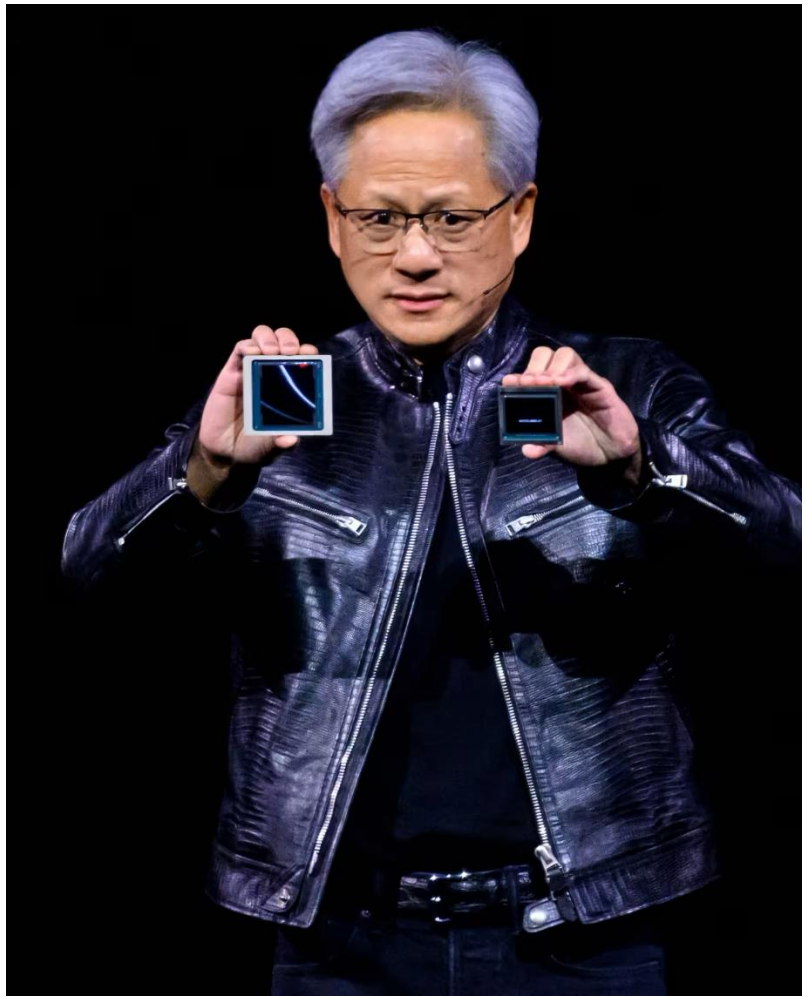


Nvidia future B200
2x larger than H100
280 Billion Transistors

*"We need **bigger GPUs**."*

Jensen Huang
Nvidia CEO
Mar 19, 2024

Bigger chips are faster and more efficient



Nvidia future B200
2x larger than H100
280 Billion Transistors

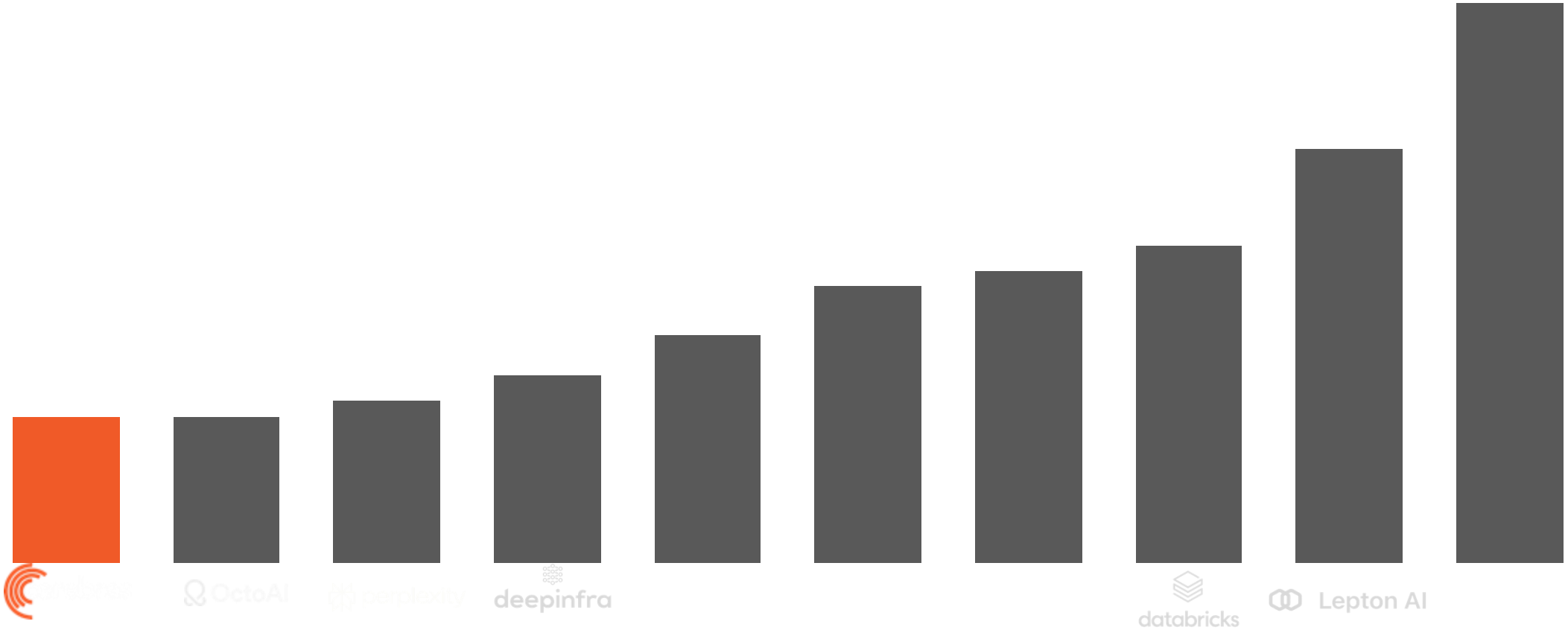
Cerebras current WSE-3
57x larger than H100
4 Trillion Transistors

>20x faster than Hyperscale Clouds

: Artificial Analysis, Llama 3.1 Results as of 9/23/24

>20x faster than Hyperscale Clouds

: Artificial Analysis, Llama 3.1 Results as of 9/23/24



Artificial Analysis

The GPU Impossible

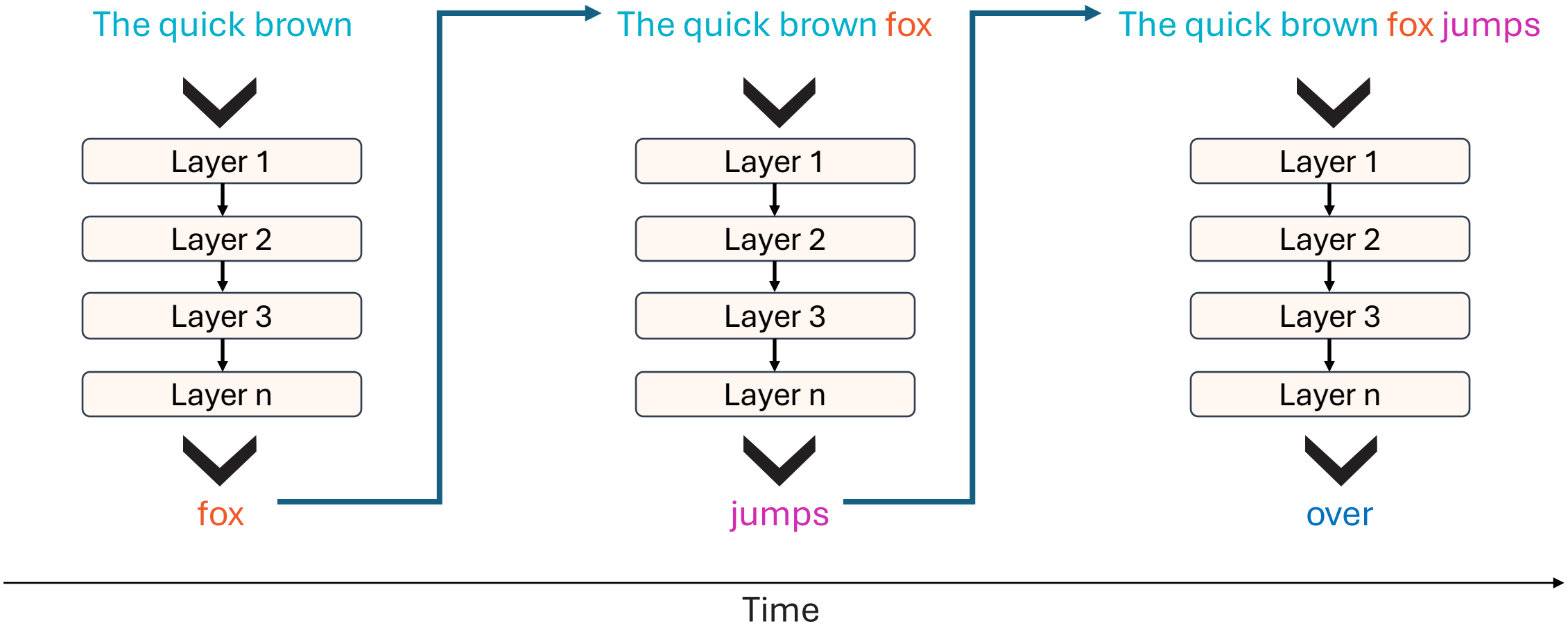
8x faster than the fastest GPU solution.

No number of GPUs can do this.

Memory Bandwidth

GenAI inference is a **memory bandwidth** problem

- Generating 1000 tokens takes 1000 serial passes through the model
- Each pass requires reading all model parameters from memory
- Low memory bandwidth is the bottleneck for generation performance



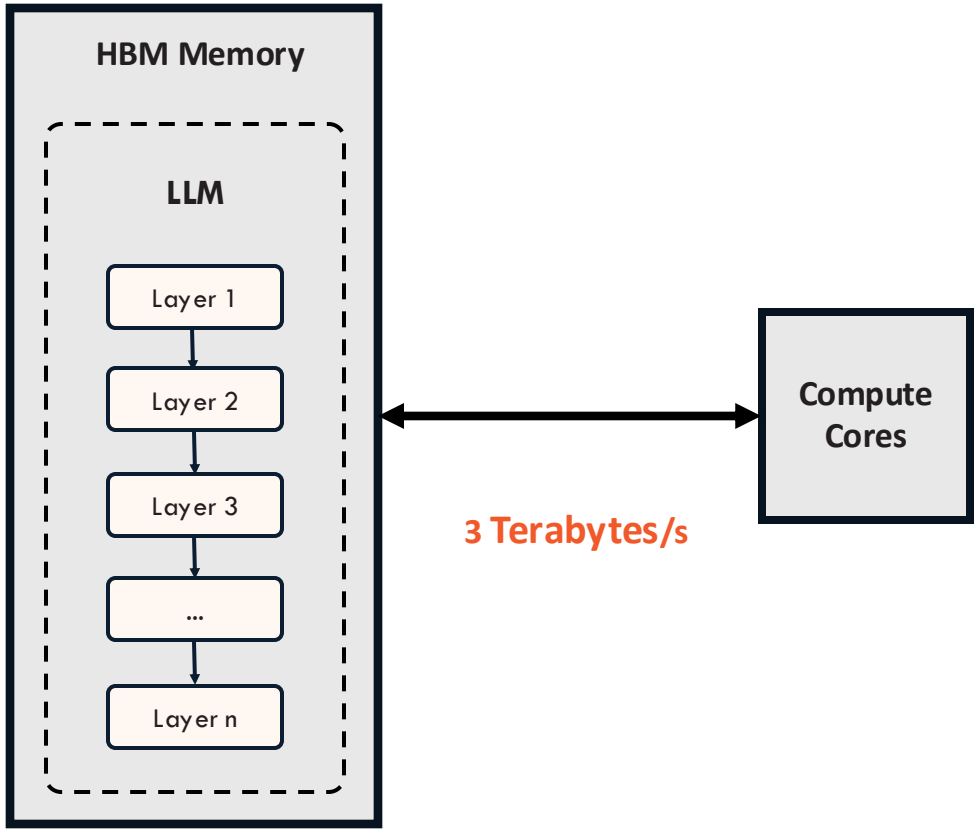
Why Wafer Scale Matters

	Cerebras WSE-3	Nvidia H100	Cerebras Advantage
Chip size	46,225 mm^2	814 mm^2	**57x**
Cores	900,000	16,896 FP32 + 528 Tensor	**52x**
On-chip memory	44 Gigabytes	0.05 Gigabytes	**880x**
Memory bandwidth	21 Petabytes/s	0.003 Petabytes/s	**7,000x**

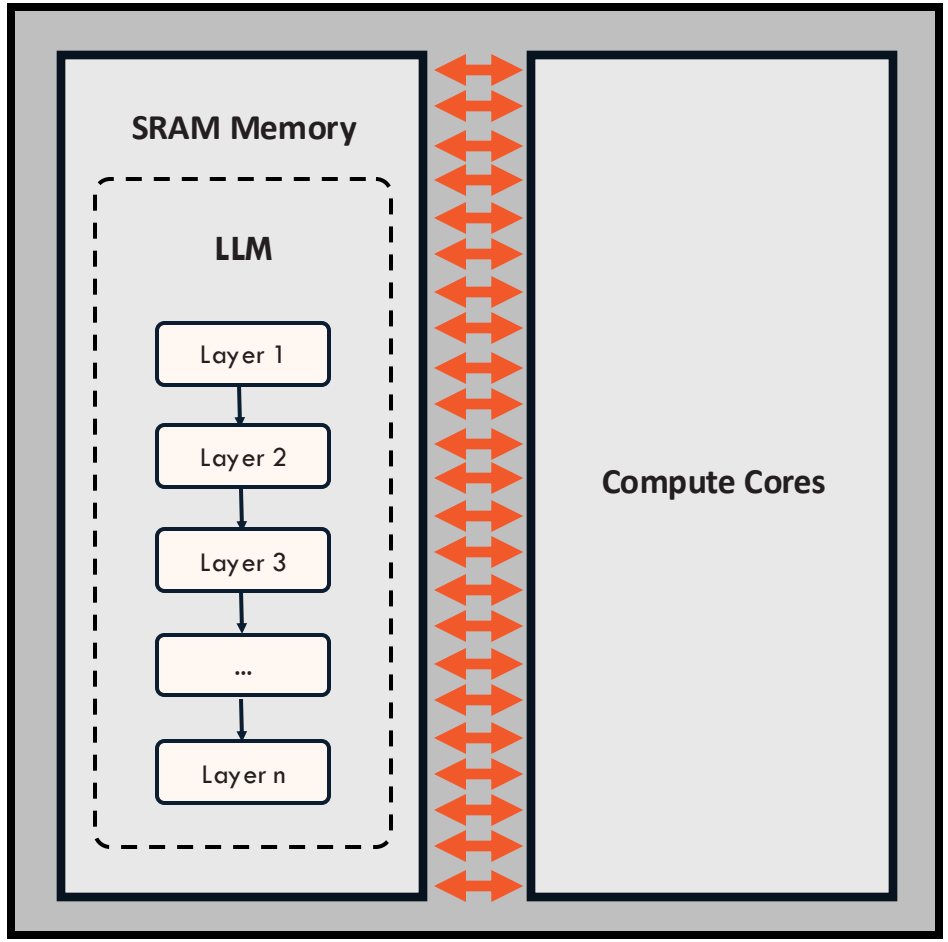
Completely removes the memory bandwidth bottleneck

Wafer Scale, SRAM based, Compute-in-Memory
7,000x more memory bandwidth than the GPU

Nvidia H100



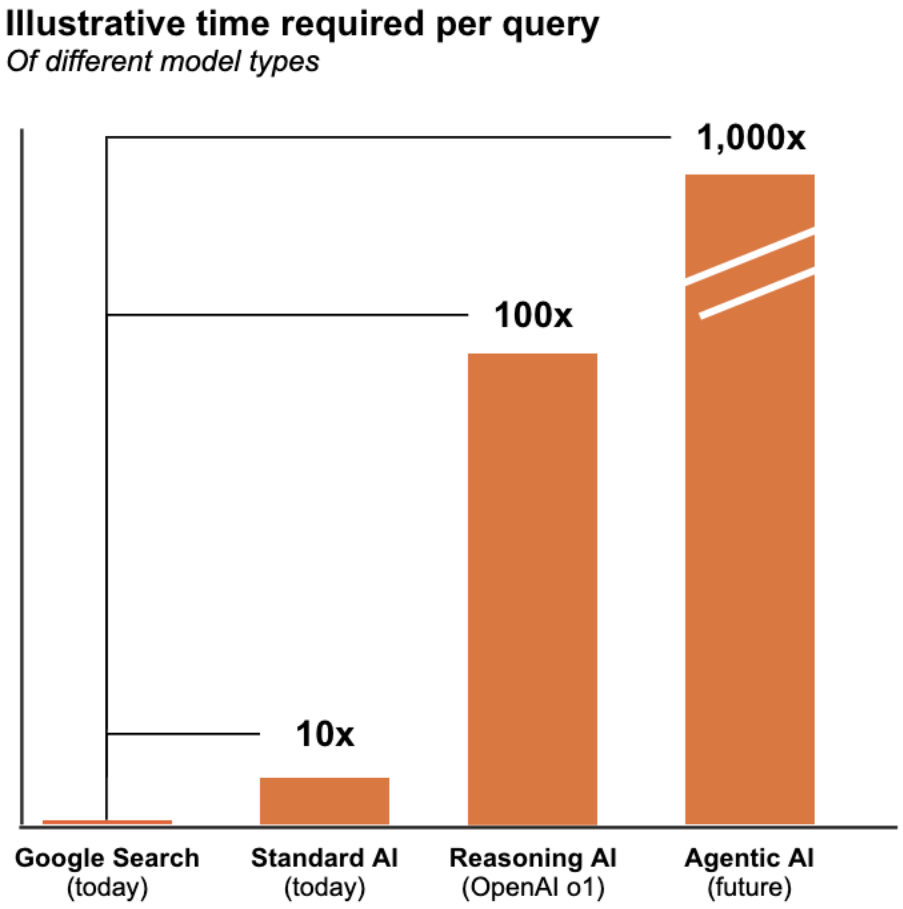
Cerebras Wafer Scale Engine 3



21 Petabytes/s

Fast Inference unlocks new capabilities

20x speed enables high user engagement and greater model capability



AI application latency limits mass adoption

Google shows that latency destroys user engagement:

- *"If results are slowed by **even a fraction of a second**, people search less."*
- *"4/5 internet **users will click away if a video stalls** while loading".*
- Search takes milliseconds, but LLMs can take seconds to tens of minutes to respond

Reasoning & Agentic flows require even more speed

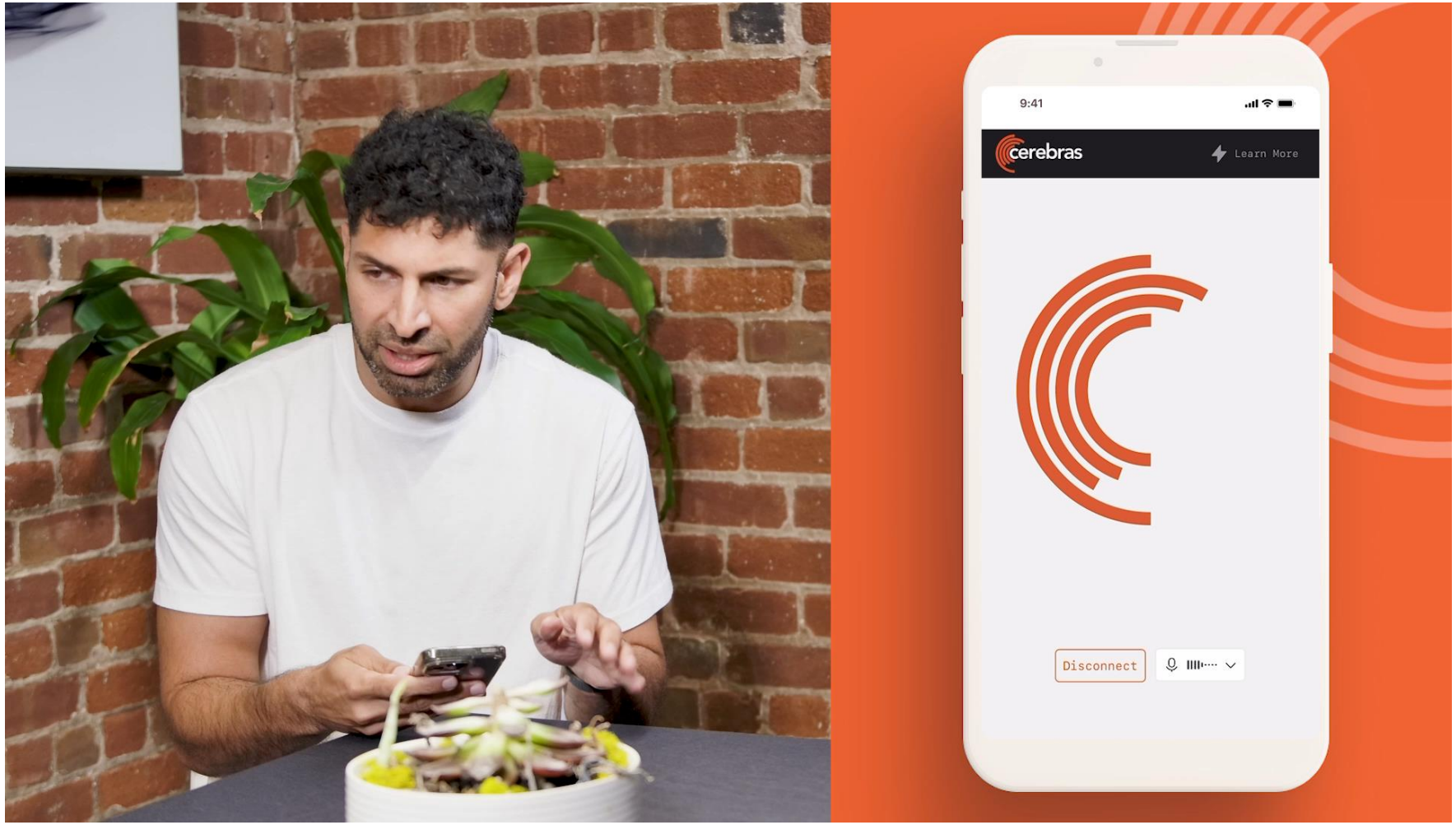
- Trading inference calls for greater chain-of-thought reasoning
- OpenAI o1 takes minutes to respond today

↑ Speed = ↑ Engagement & Capability = ↑ ROI

Source: Our World In Data "Computation used to train notable artificial intelligence systems". OpenAI "Learning to reason with LLMs".

More Speed = More Engagement

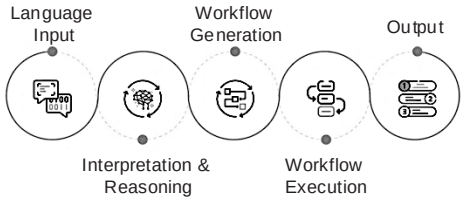
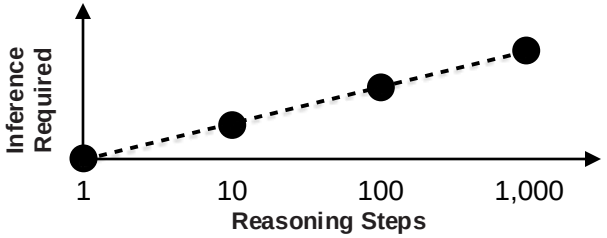
Unlock real-time interactivity.



Cerebras **cuts inference time by 10x** and enables economic scaling of AI applications

Application		GPUs	cerebras
Generative text & search *Inference queries take **10x longer** than traditional search queries*	Q: “Where to visit in Italy?”	5 seconds	**0.5 seconds**
Generative voice & video *Require **<100ms** latency*		700 milliseconds	**70 milliseconds**

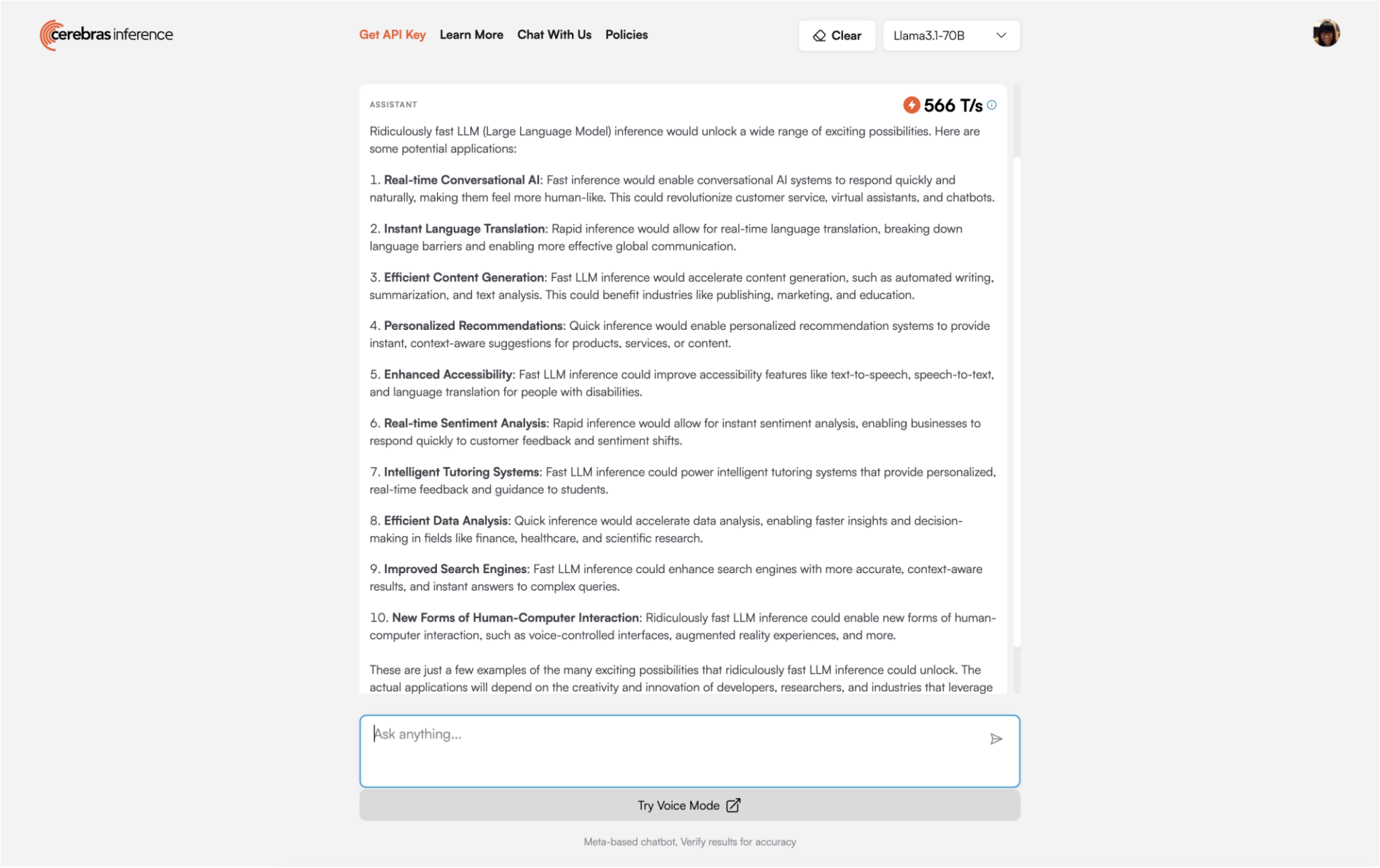
Speed is critical in existing ▲ and emerging ▼ applications. Google has shown that search results need to return in milliseconds to capitalize on user engagement.

AI agents *Can think **10+ minutes** per query*		10 minutes	**1 minute**
Reasoning models *Those like OpenAI o1 conduct **100x more inference** per query*		10 reasoning steps	**100 reasoning steps**

Time is money. Economic scaling of latency-intensive AI applications **requires a fundamentally higher-speed solution**.

Inference Offering Summary

Experience Cerebras Inference speeds at inference.cerebras.ai



Experiment with models in our Developer Playground

cerebras

- Playground
- API Keys
- Usage & Limits
- Pricing
- Settings
- Documentation
- Chat Demo
- Why Cerebras?
- Say Hello!

View code | Clear | Llama3.1-70B

SYSTEM

Please answer as if you are an expert at model inferencing

USER

Why is high memory bandwidth so important to fast GenAI model inferencing?

ASSISTANT

452 T/s

As an expert in model inferencing, I can tell you that high memory bandwidth is crucial for fast GenAI (Generative Artificial Intelligence) model inferencing because of the way these models are designed and the computations they perform.

GenAI models, such as transformers and LSTMs, rely heavily on matrix multiplications and tensor operations to process input data and generate output. These operations involve accessing and manipulating large amounts of data in memory, which requires high memory bandwidth to perform efficiently.

Here are some reasons why high memory bandwidth is essential for fast GenAI model inferencing:

1. **Matrix multiplications**: GenAI models perform numerous matrix multiplications, which involve accessing and multiplying large matrices. High memory bandwidth allows for faster access to these matrices, reducing the time it takes

Message Cerebras

User | Assistant | Add | Run

Temperature 0.5

Max tokens 512

Top P 0.5

Seed 1

Stream

JSON

Stop sequence

Switch to Cerebras Inference in 30 seconds with OpenAI-compatible API

```python
import os
from openai import OpenAI

client = OpenAI(
  api_key=os.environ.get("OPENAI_API_KEY"),
)

response = client.chat.completions.create(
  model="gpt-3.5-turbo",
  messages=[...],
  temperature=1,
  max_tokens=256,
  top_p=1,
  frequency_penalty=0,
  presence_penalty=0
)
```

```python
import os
from openai import OpenAI

client = OpenAI(
  base_url="https://api.cerebras.ai/v1",
  api_key=os.environ.get("CEREBRAS_API_KEY"),
)

response = client.chat.completions.create(
  model="llama3.1-70b",
  messages=[...],
  temperature=1,
  max_tokens=256,
  top_p=1,
  frequency_penalty=0,
  presence_penalty=0
)
```

Features

Function calling

Json output

Coming soon

Structured output

Cerebras Inference Solutions

Pay-As-You-Go

- Serverless API endpoint charged by millions of tokens
- Easily scale up or down
- No SLA. Rates and token limits subject to change
- Ideal for early-stage exploration and experimentation

Provisioned Throughput

- Serverless API endpoint with guaranteed performance
- Fine-tuned model support
- Fixed monthly or yearly charge
- Minimum commitment
- Ideal for startups with consistent, predictable workloads

Private System via Cerebras Cloud

- Private deployment managed by Cerebras
- Full control over data residency
- Maintained by Cerebras
- Ideal for sustained workloads of digital-native enterprises

Private System on Premises

- Deployment in customer's data center
- Customer fully owns infrastructure and information
- Hardware optimization for specific workloads
- Long-term cost savings
- Ideal for large enterprise, government, healthcare & secure AI solutions

Serverless endpoint

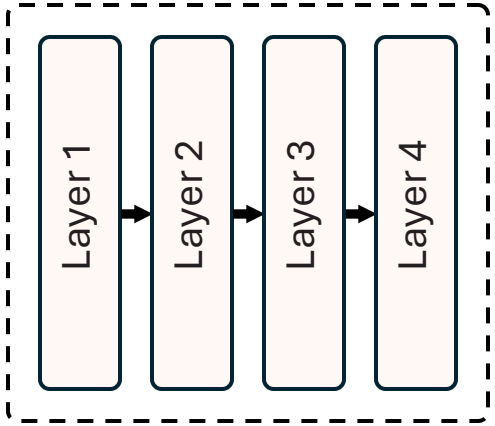
Private deployment

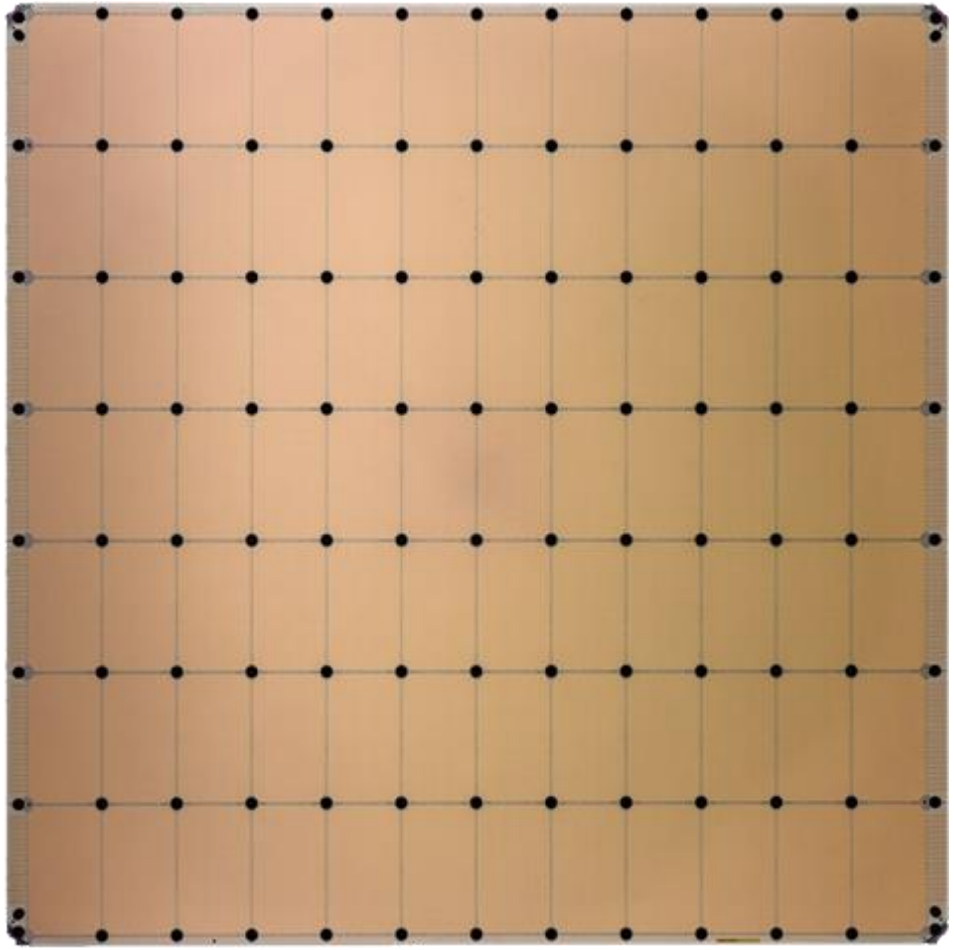
Lowest Latency Generation

Pipeline execution on a single chip

Massive memory bandwidth enables opposite execution model

- GPU: multiple chips to run a single layer
- Cerebras: fraction of a chip to run a single layer





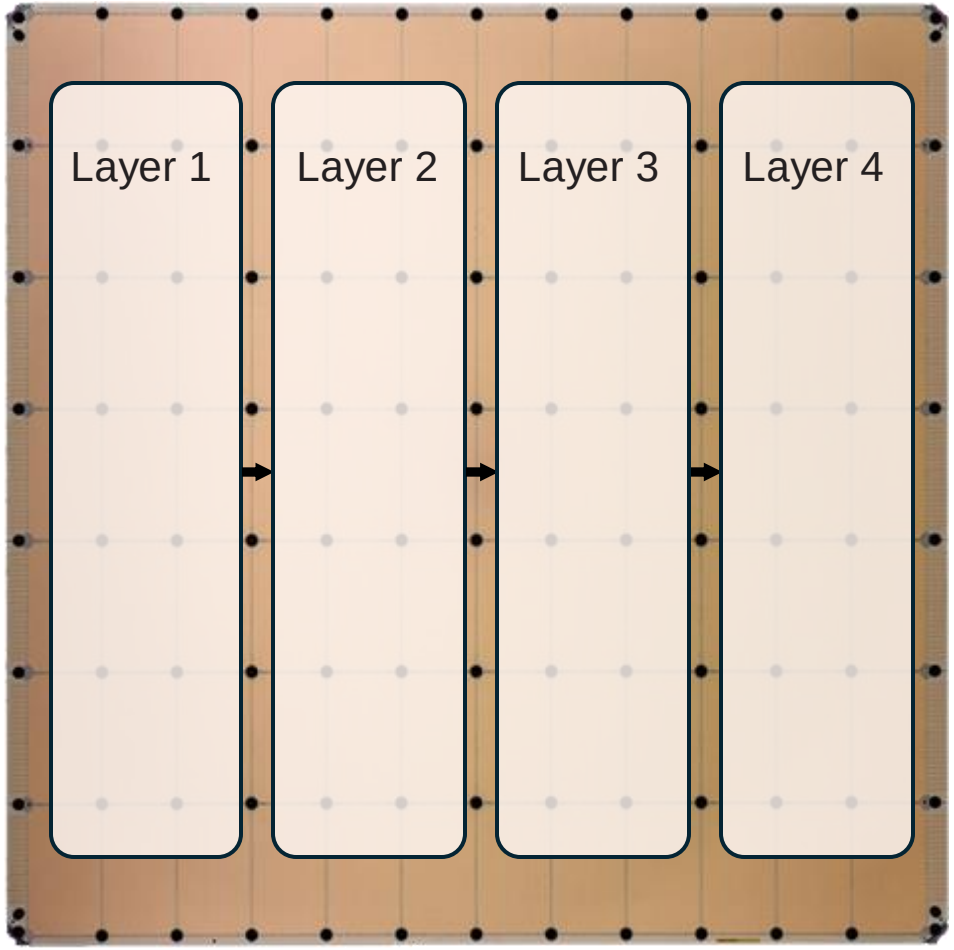
Pipeline execution on a single chip

Massive memory bandwidth enables opposite execution model

- GPU: multiple chips to run a single layer
- Cerebras: fraction of a chip to run a single layer

Pipeline mapping of model

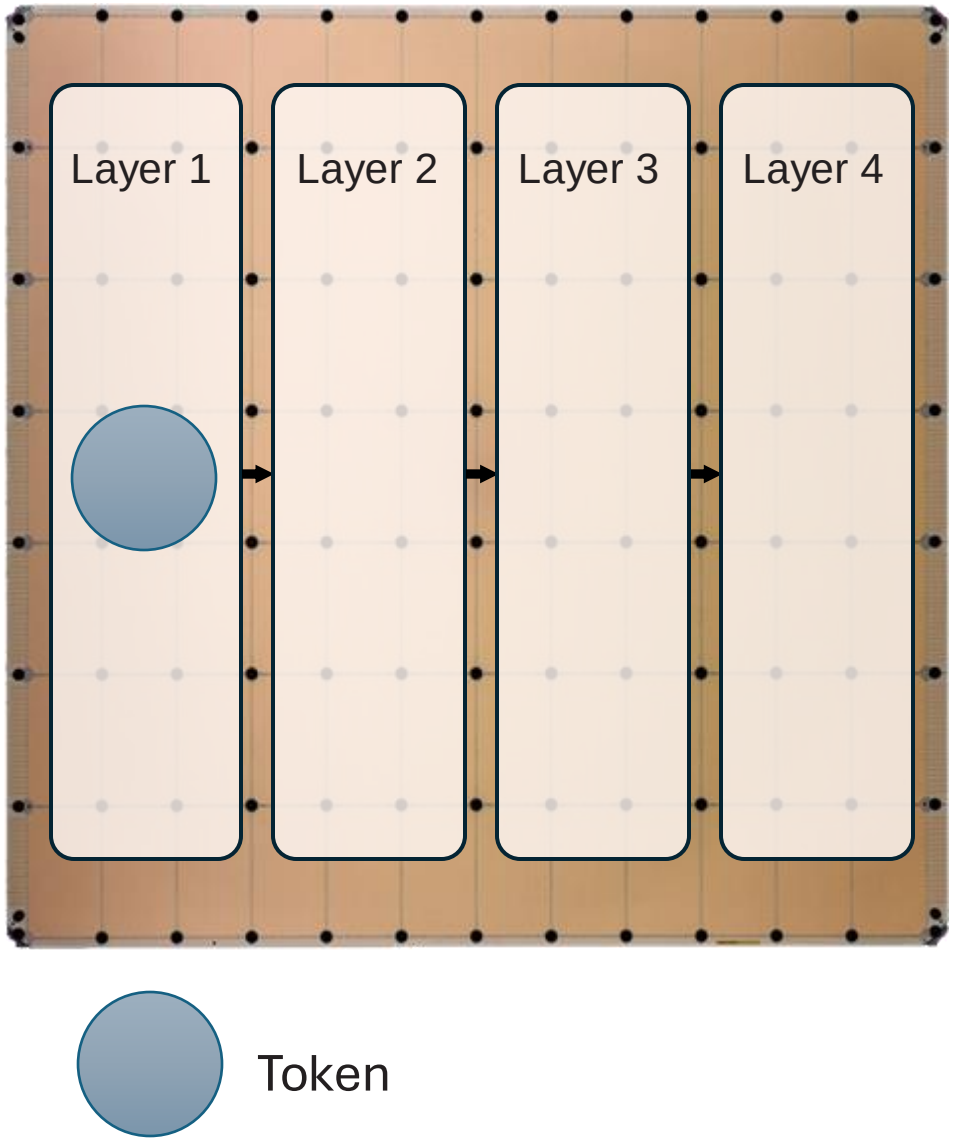
- Model layers are mapped to wafer regions
- Size of wafer region determined based on memory and compute requirements
- Model weights and KV cache stored locally in the region memory close to the compute



Pipeline execution

A low latency pipeline

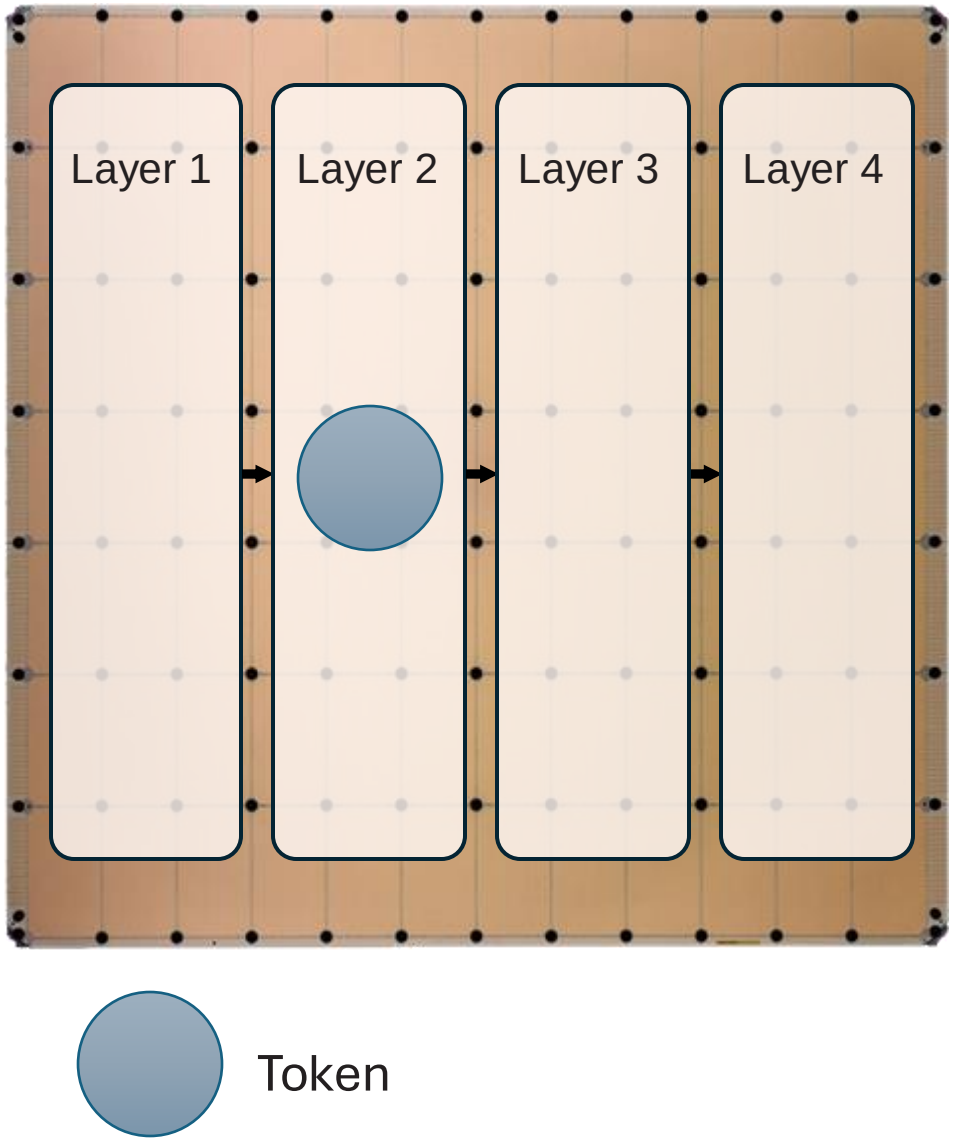
- Each wafer region processes 1 token
  - Enough memory bandwidth to run local batch size 1
  - Memory to feed compute datapath at full speed
  - Enabling optimized performance for matrix*vector
  - Local fabric interconnect to maintain low latency



Pipeline execution

A low latency pipeline

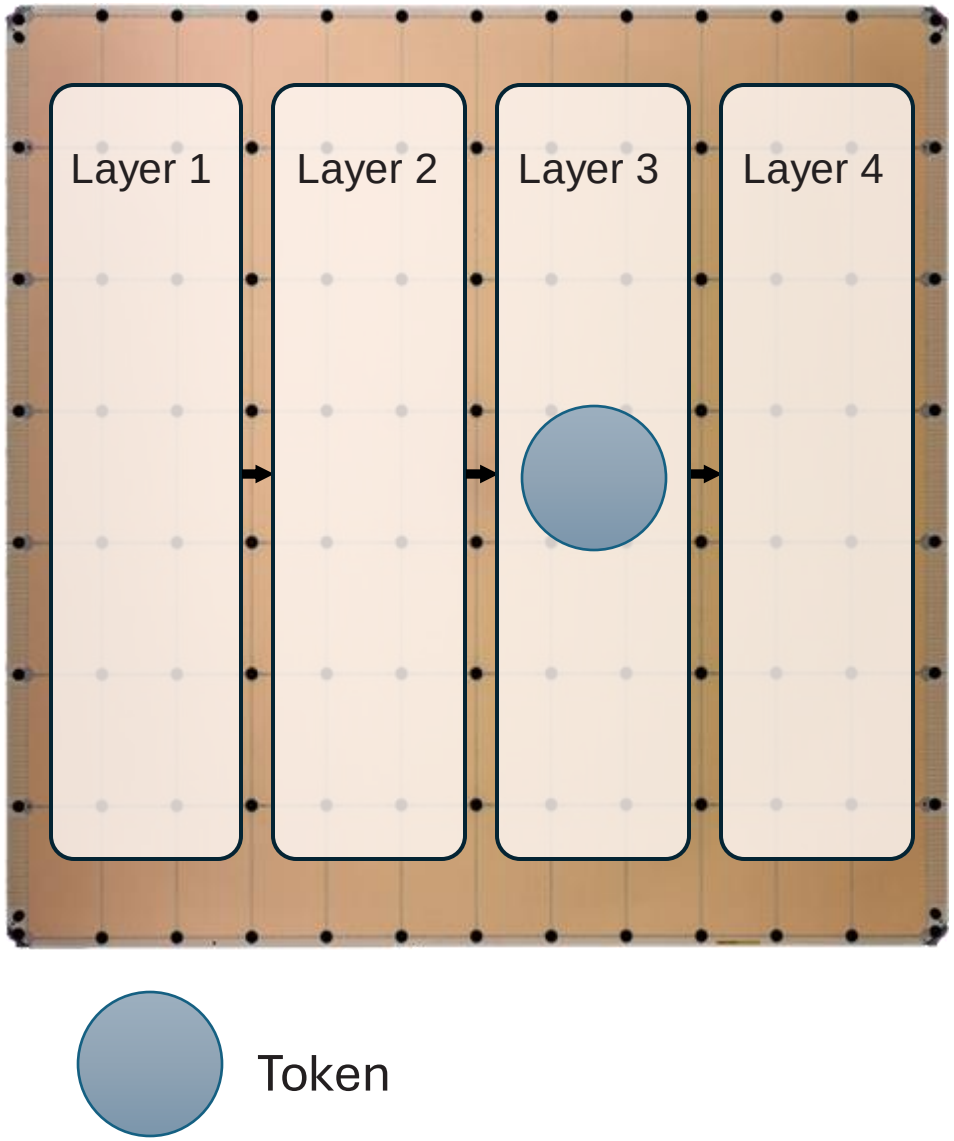
- Each wafer region processes 1 token
  - Enough memory bandwidth to run local batch size 1
  - Memory to feed compute datapath at full speed
  - Enabling optimized performance for matrix*vector
  - Local fabric interconnect to maintain low latency
- Regions are placed to reduce latency
  - The next region is physically adjacent
  - Virtually no latency between pipe stages
  - Possible because it's all done on-chip



Pipeline execution

A low latency pipeline

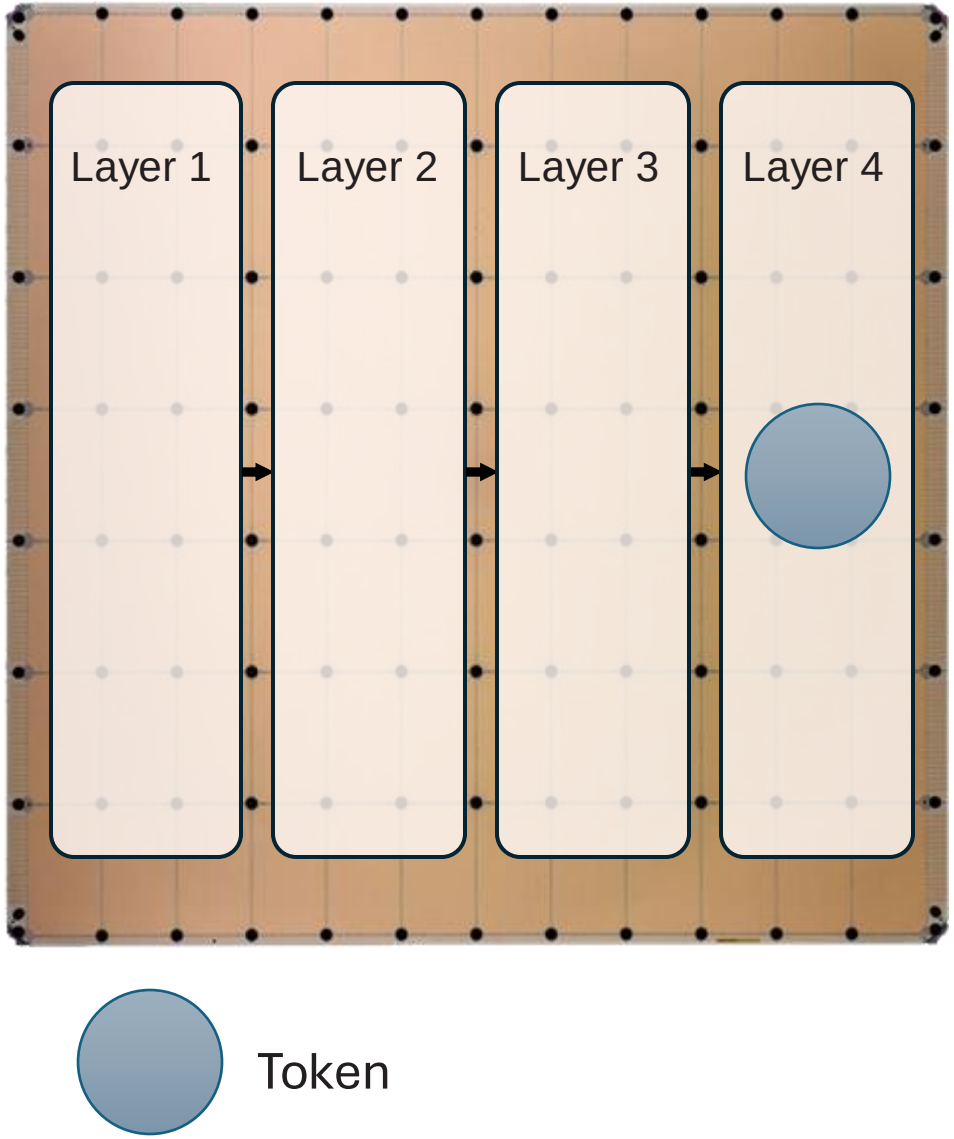
- Each wafer region processes 1 token
  - Enough memory bandwidth to run local batch size 1
  - Memory to feed compute datapath at full speed
  - Enabling optimized performance for matrix*vector
  - Local fabric interconnect to maintain low latency
- Regions are placed to reduce latency
  - The next region is physically adjacent
  - Virtually no latency between pipe stages
  - Possible because it's all done on-chip



Pipeline execution

A low latency pipeline

- Each wafer region processes 1 token
  - Enough memory bandwidth to run local batch size 1
  - Memory to feed compute datapath at full speed
  - Enabling optimized performance for matrix*vector
  - Local fabric interconnect to maintain low latency
- Regions are placed to reduce latency
  - The next region is physically adjacent
  - Virtually no latency between pipe stages
  - Possible because it's all done on-chip

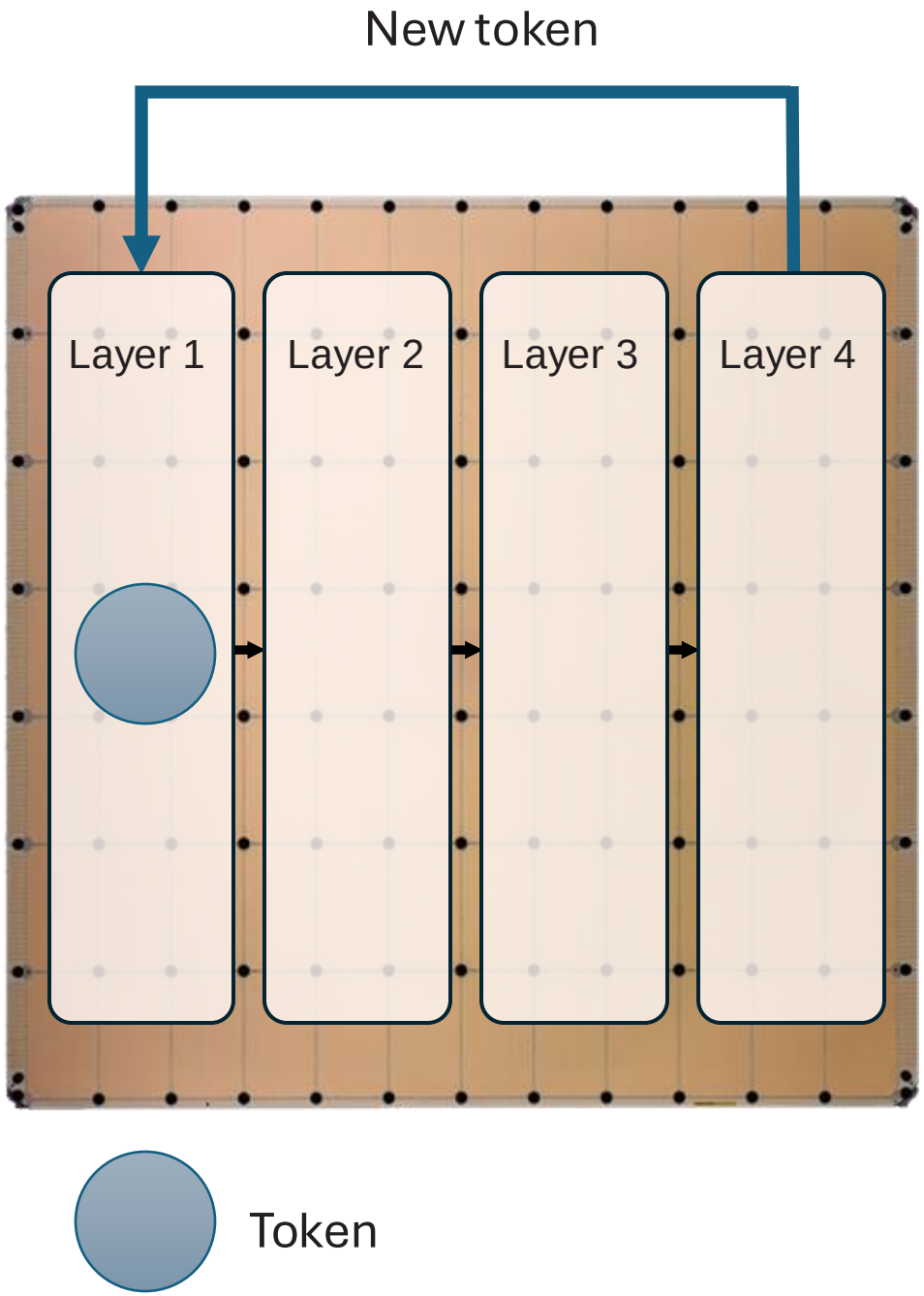


Pipeline execution

A low latency pipeline

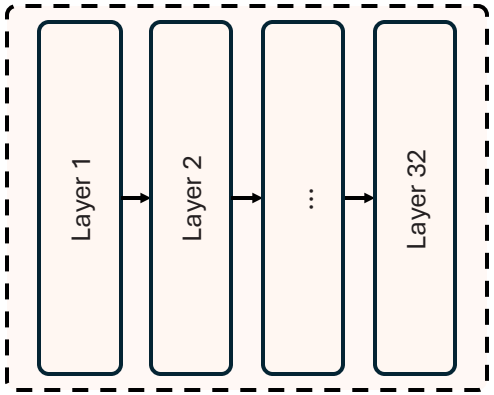
- Each wafer region processes 1 token
  - Enough memory bandwidth to run local batch size 1
  - Memory to feed compute datapath at full speed
  - Enabling optimized performance for matrix*vector
  - Local fabric interconnect to maintain low latency
- Regions are placed to reduce latency
  - The next region is physically adjacent
  - Virtually no latency between pipe stages
  - Possible because it's all done on-chip
- Last region output cycled back to generate next token

Pipelined execution enables super-fast token generation

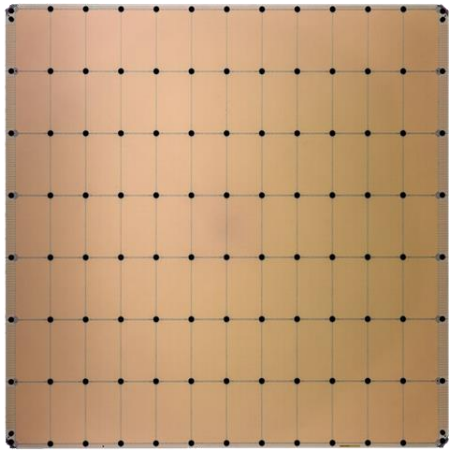


Scalability

Large models fit on our large on-chip memory



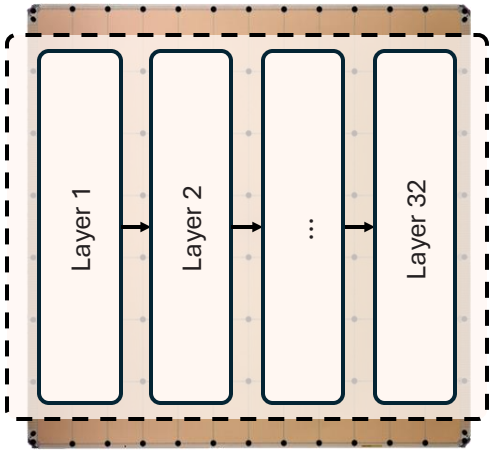
Llama3.1-8B
8 billion parameters
16 GB of memory (FP16)



WSE-3
44GB of SRAM

Large models fit on our large on-chip memory

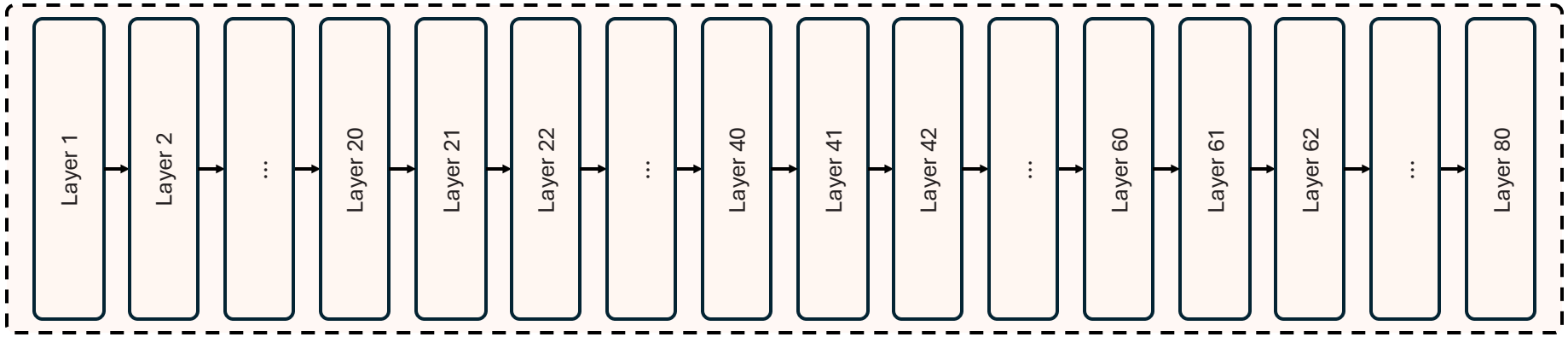
- Models that fit entirely on the wafer are mapped directly



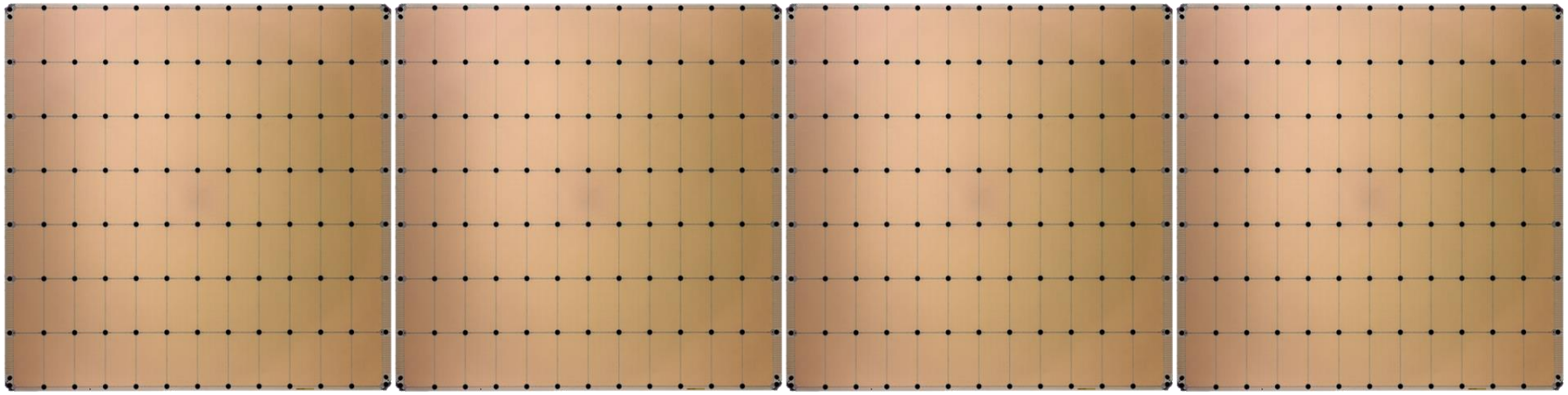
Llama3.1-8B
8 billion parameters
16 GB of memory (FP16)

WSE-3
44GB of SRAM

Naturally scales to multiple wafers



Llama3.1-70B
70 billion parameters
140 GB of memory (FP16)



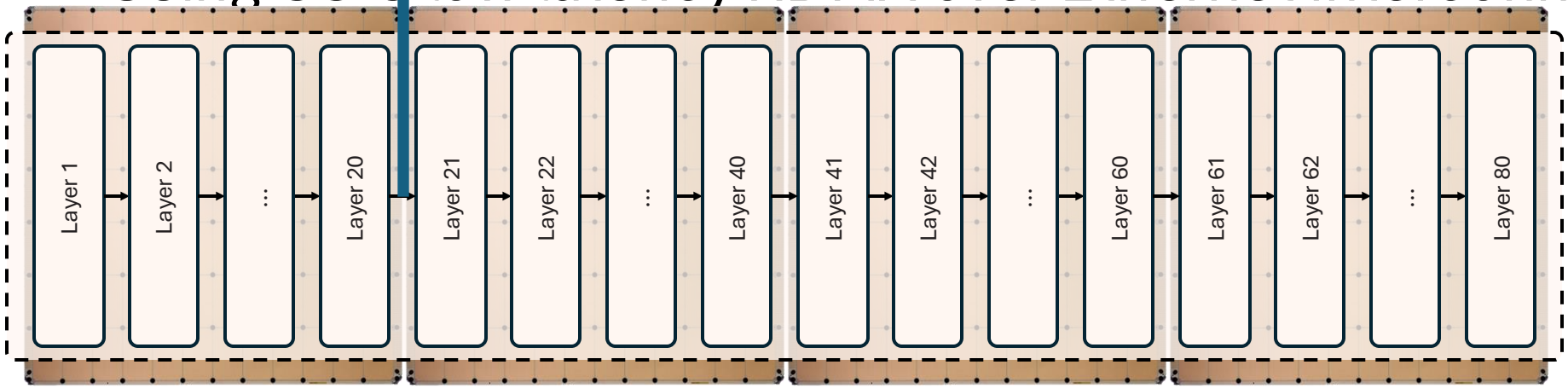
4x WSE-3
176GB of SRAM

Naturally scales to multiple wafers

- Models that require more memory are mapped to multiple wafers
- Keep almost all high communication on wafer, using internal high bandwidth fabric
- Transfer only activations between wafers, requiring relatively lower bandwidth
- Using CS-3 low latency RDMA over Ethernet interconnect between

Latency:
25-300 ns <5us
4 hops required
<1% impact to latency

Bandwidth:
CS-3 has 1.2Tbps
100Gbps required
<10% of available



Llama3.1-70B
70 billion parameters
140 GB of memory (FP16)

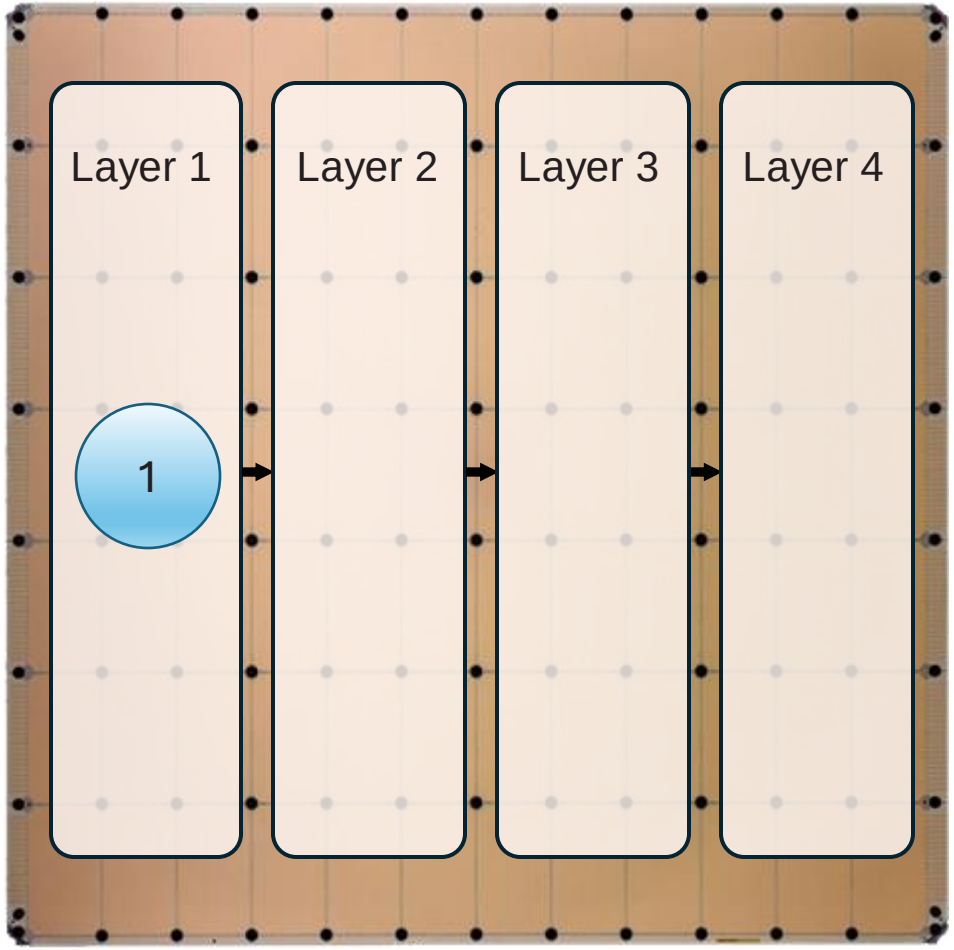
4x WSE-3
176GB of SRAM

High Throughput

Cerebras enables **low latency and high throughput**

More than enough memory bandwidth for single user enables high multi-user throughput

- Single user is using only a fraction of the total memory bandwidth
- We can use the extra bandwidth to support multiple users

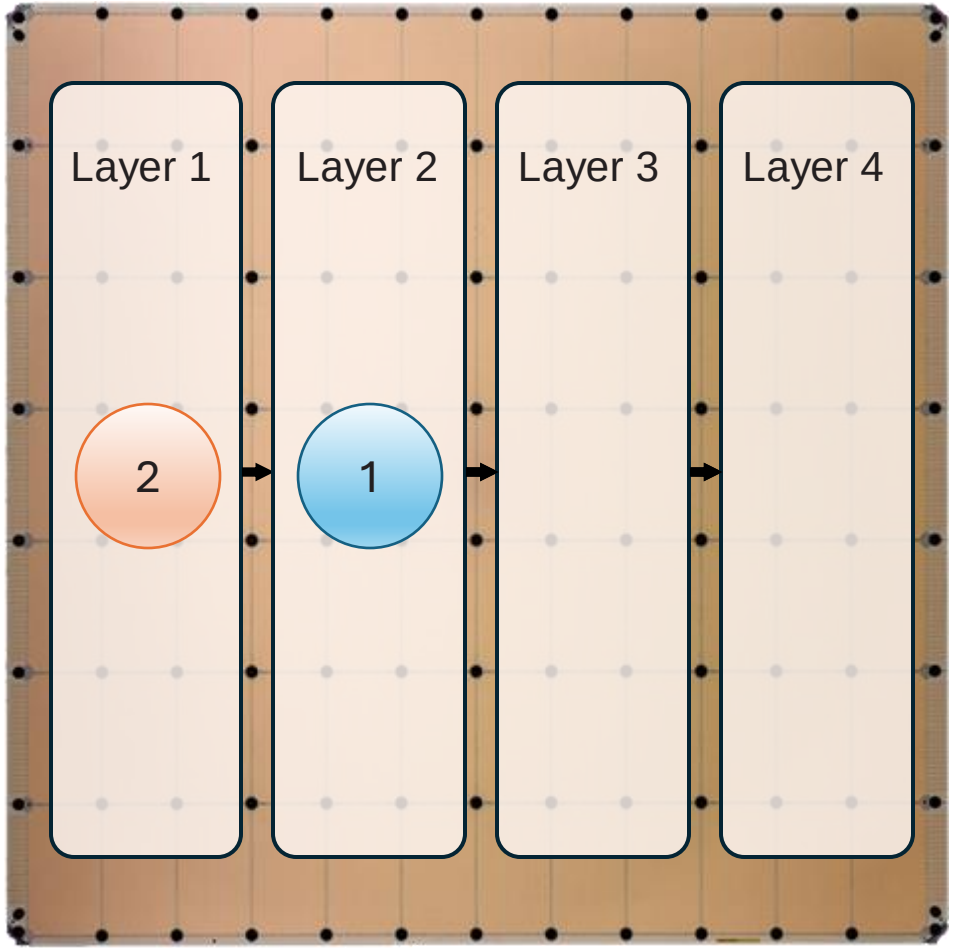


Cerebras enables **low latency and high throughput**

More than enough memory bandwidth for single user enables high multi-user throughput

- Single user is using only a fraction of the total memory bandwidth
- We can use the extra bandwidth to support multiple users

- Additional users all run in parallel
- Each accessing the model simultaneously

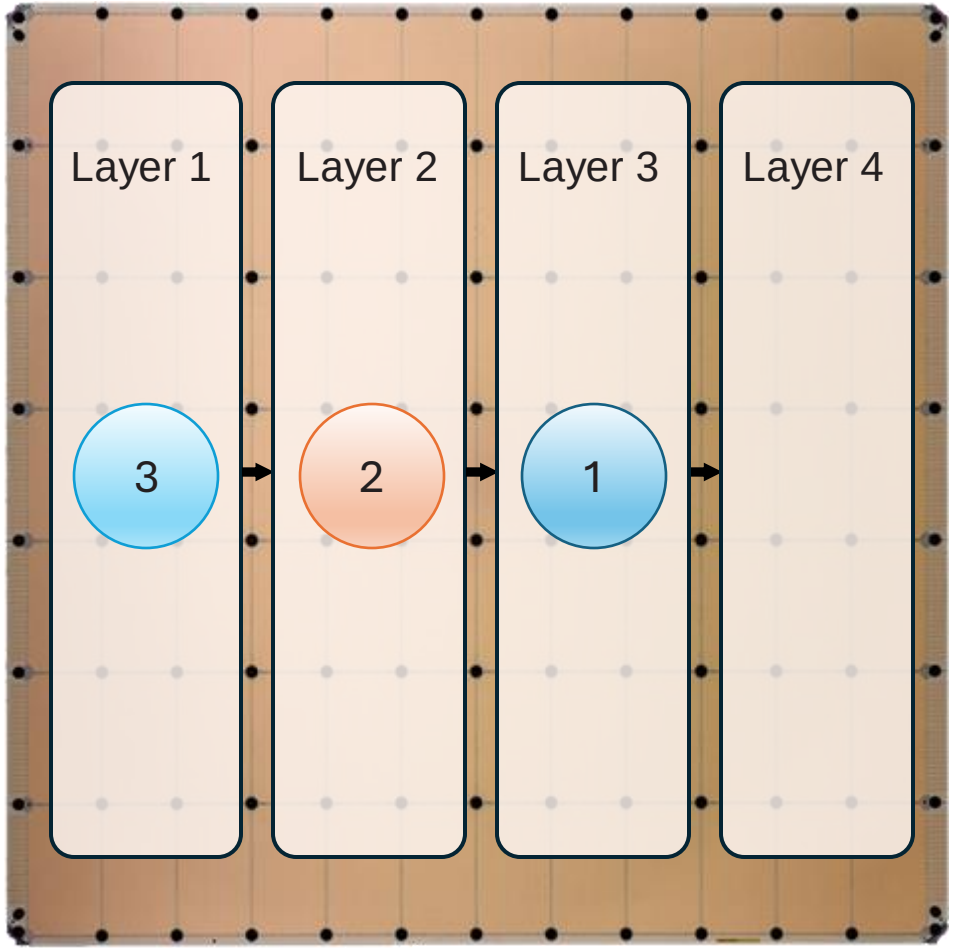


Cerebras enables **low latency and high throughput**

More than enough memory bandwidth for single user enables high multi-user throughput

- Single user is using only a fraction of the total memory bandwidth
- We can use the extra bandwidth to support multiple users

- Additional users all run in parallel
- Each accessing the model simultaneously
- Every user gets full performance
- All pipe stages to run at the same time



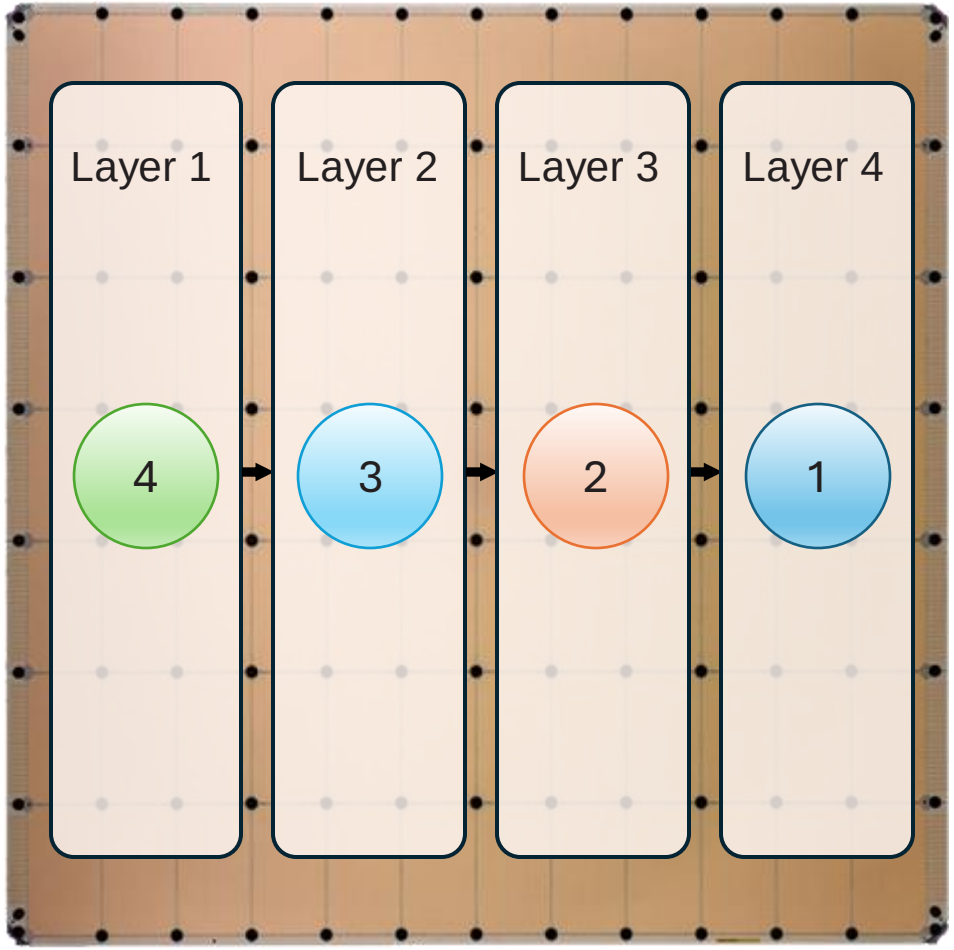
Cerebras enables **low latency and high throughput**

More than enough memory bandwidth for single user enables high multi-user throughput

- Single user is using only a fraction of the total memory bandwidth
- We can use the extra bandwidth to support multiple users

- Additional users all run in parallel
- Each accessing the model simultaneously
- Every user gets full performance
- All pipe stages to run at the same time

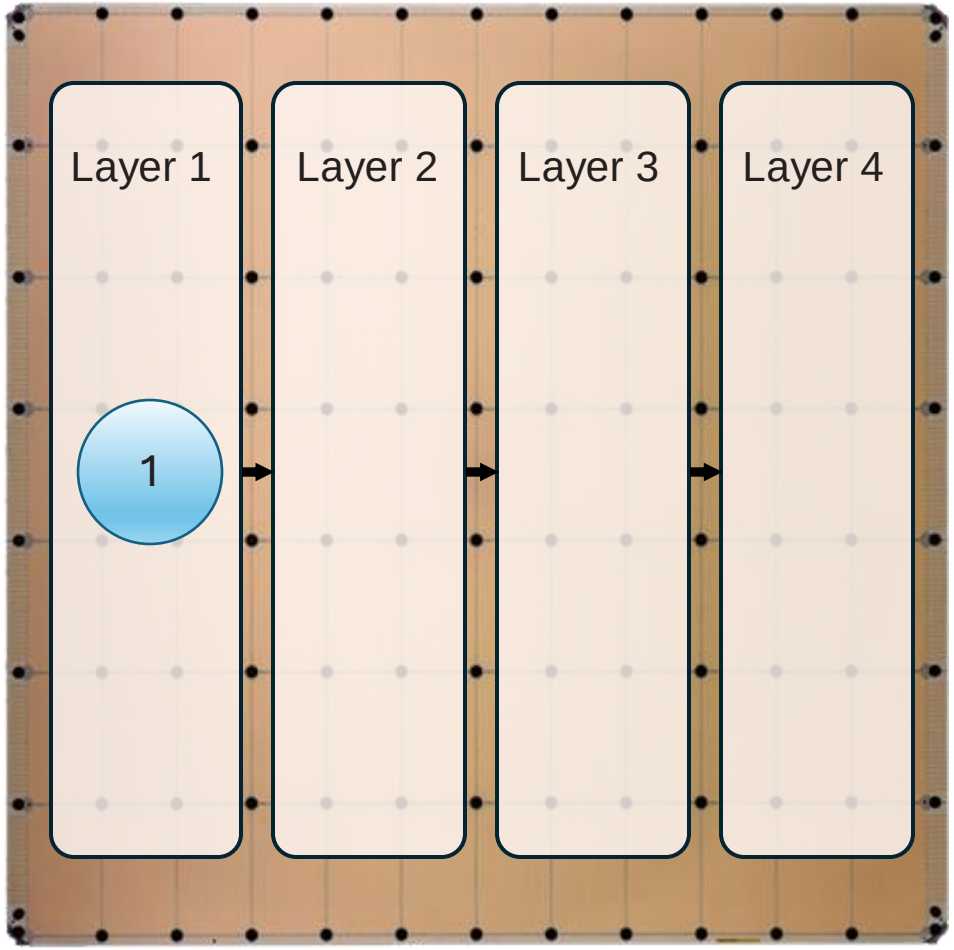
Full pipeline model parallelism on single chip



High performance input processing

Input tokens can be processed in parallel

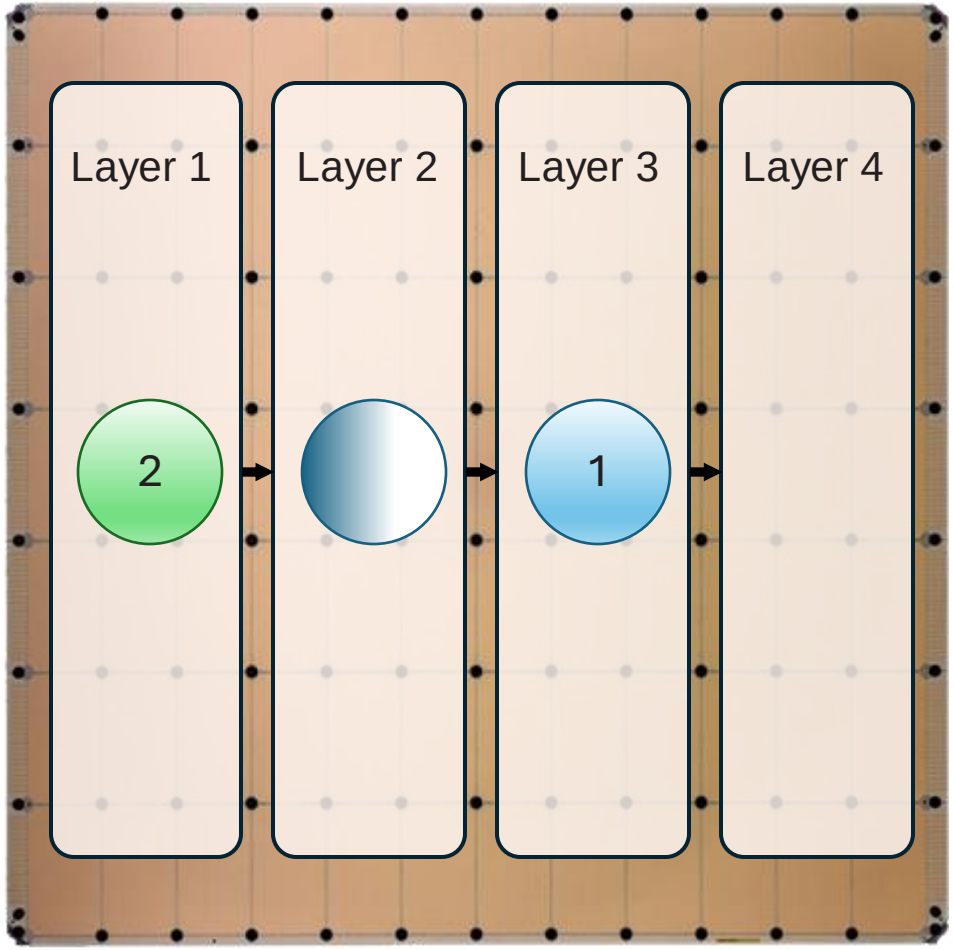
- Input is known upfront so no need to wait for full pipeline execution for next token



High performance input processing

Input tokens can be processed in parallel

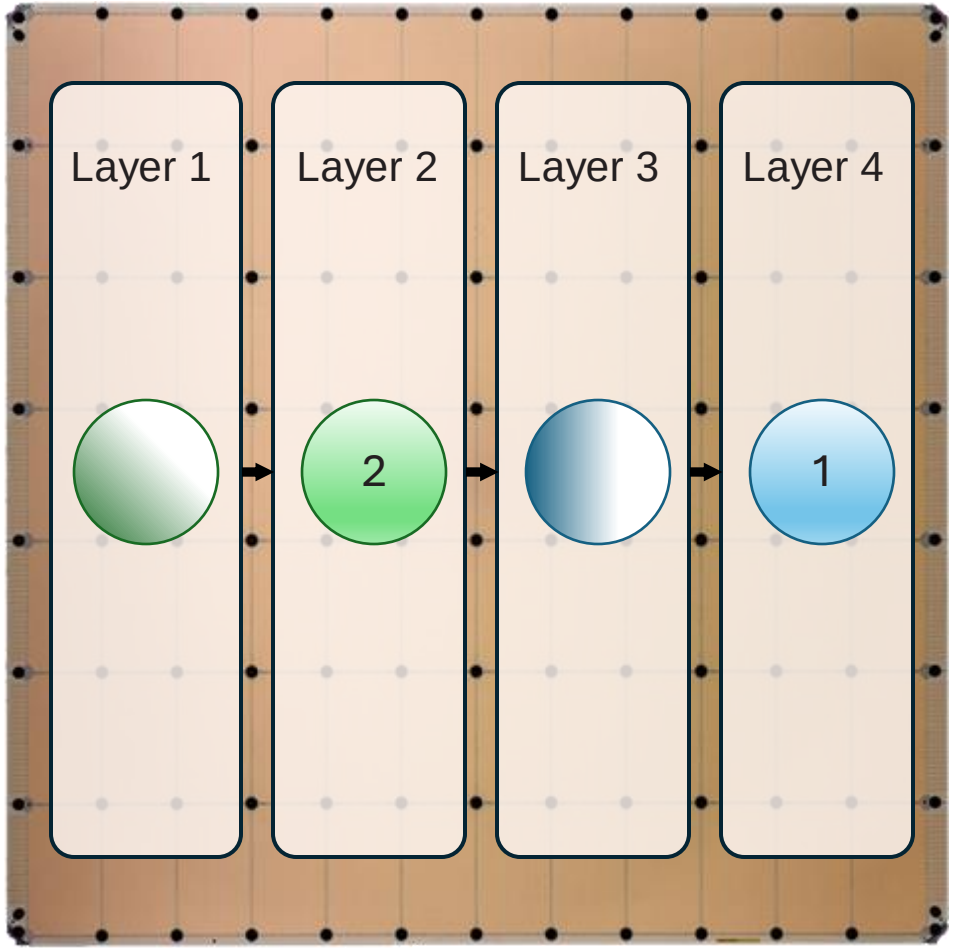
- Input is known upfront so no need to wait for full pipeline execution for next token
- Run multiple single user input tokens in multiple pipe stages
- Can achieve higher single user input speed using empty pipe stages when fewer users



High performance input processing

Input tokens can be processed in parallel

- Input is known upfront so no need to wait for full pipeline execution for next token
- Run multiple single user input tokens in multiple pipe stages
- Can achieve higher single user input speed using empty pipe stages when fewer users



Flexible faster input processing driving maximum throughput

Inference Cluster Architecture

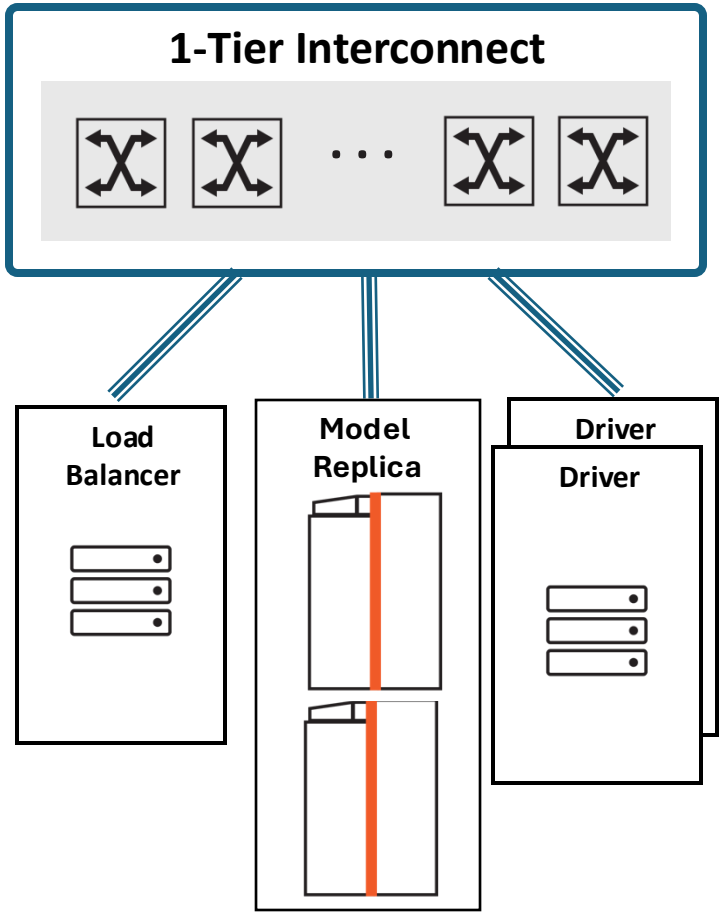
Cerebras Private Cloud

Key components:

- **Cerebras CS-3 Systems**: CS-3 systems form model replicas, each serving a single model instance with specific throughput. Multiple CS-3 systems may be used within a single model replica to meet performance requirements.
  - **Scalability**: Multiple model replicas can be combined in a single cluster
- **Supporting Infrastructure**: Additional servers perform load balancing and replica management functions. Switches provide interconnectivity.

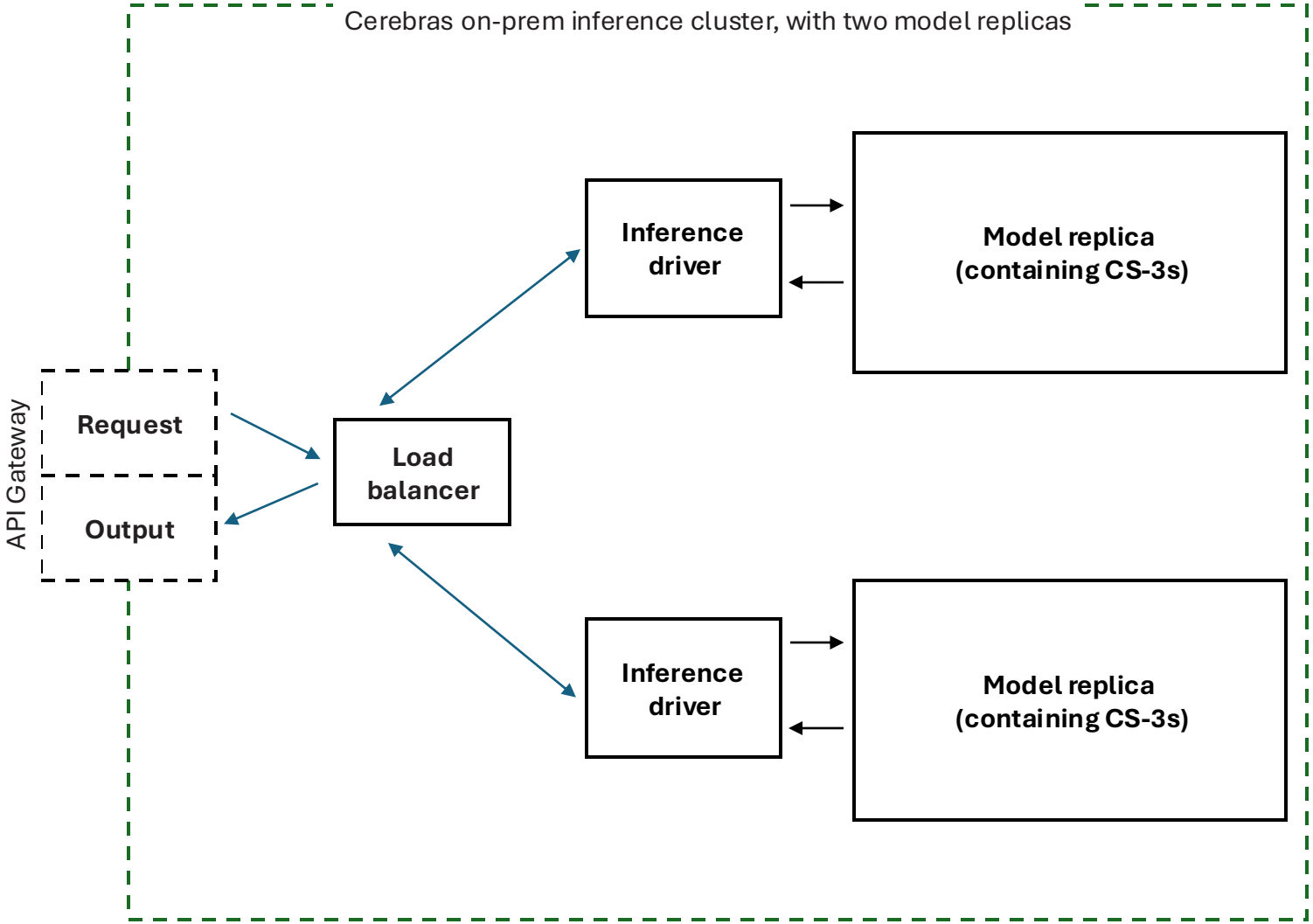
Deployed as a Complete Appliance: Cerebras designs, installs, and deploys the full solution, with warranty and support included.

Cerebras Inference Private Cloud, Architecture



Inference Architecture Overview

- API gateway & load balancer (on a CPU server)
  - Accepts and authenticates API requests
  - Handles rate limiting functionality
  - Routes requests to the most available model replica
  - Returns API response to user
- Inference drivers (on a CPU server)
  - Point of entry to the model replica
  - Loads model weights at startup
- Core inference systems (on CS-3s)
  - Systems perform autoregressive inference in a loop

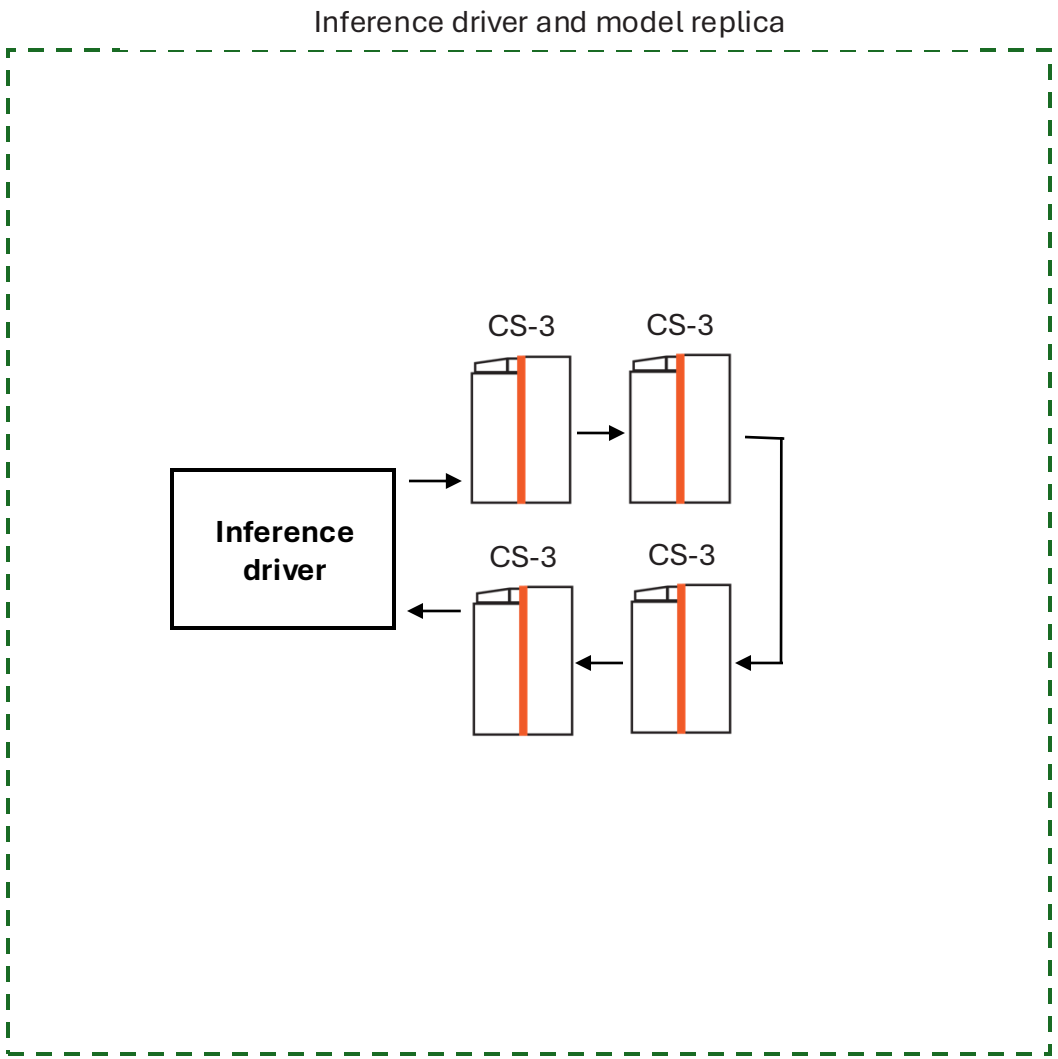


Core Inference System – Model Replica Deep Dive

Each inference replica includes the following:

- Core inference driver servers (on CPUs)
  - Loads model weights to CS-3s
  - Interfaces between API and accelerators for requests and outputs

- Core inference accelerator(s) on CS-3s
  - Model weights are distributed across one or more CS-3s
  - The number of CS-3s per replica depends on the characteristics and size of the model
  - Systems perform autoregressive inference in a loop



Training Deep Dive

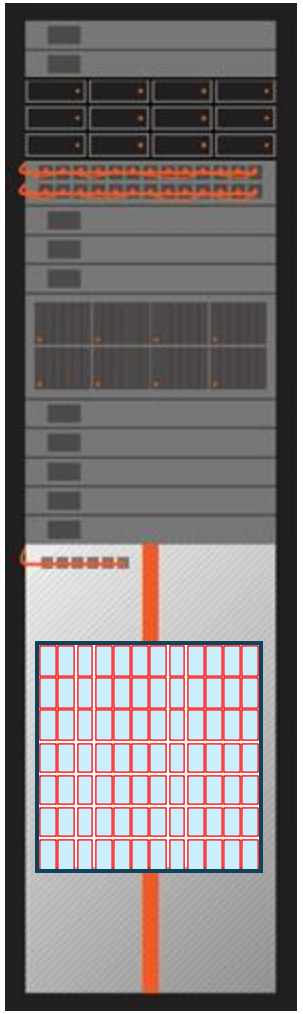
CS-3 vs. GPU

Orders of Magnitude Performance Advantage

	Cerebras CS-3	Nvidia H100	Cerebras Advantage
Chip size	46,225 mm^2	814 mm^2	57x
Cores	900,000	16,896 FP32 + 528 Tensor	52x
On-chip memory	44 Gigabytes	0.05 Gigabytes	880x
Memory bandwidth	21 Petabytes/sec	0.003 Petabytes/sec	7,000X
Fabric bandwidth	214 Petabits/sec	0.0576 Petabits/sec	3,715X

Enabling large scale training
Finetune LLaMA 70B on 1B tokens in a day
<u>on a single chip</u>

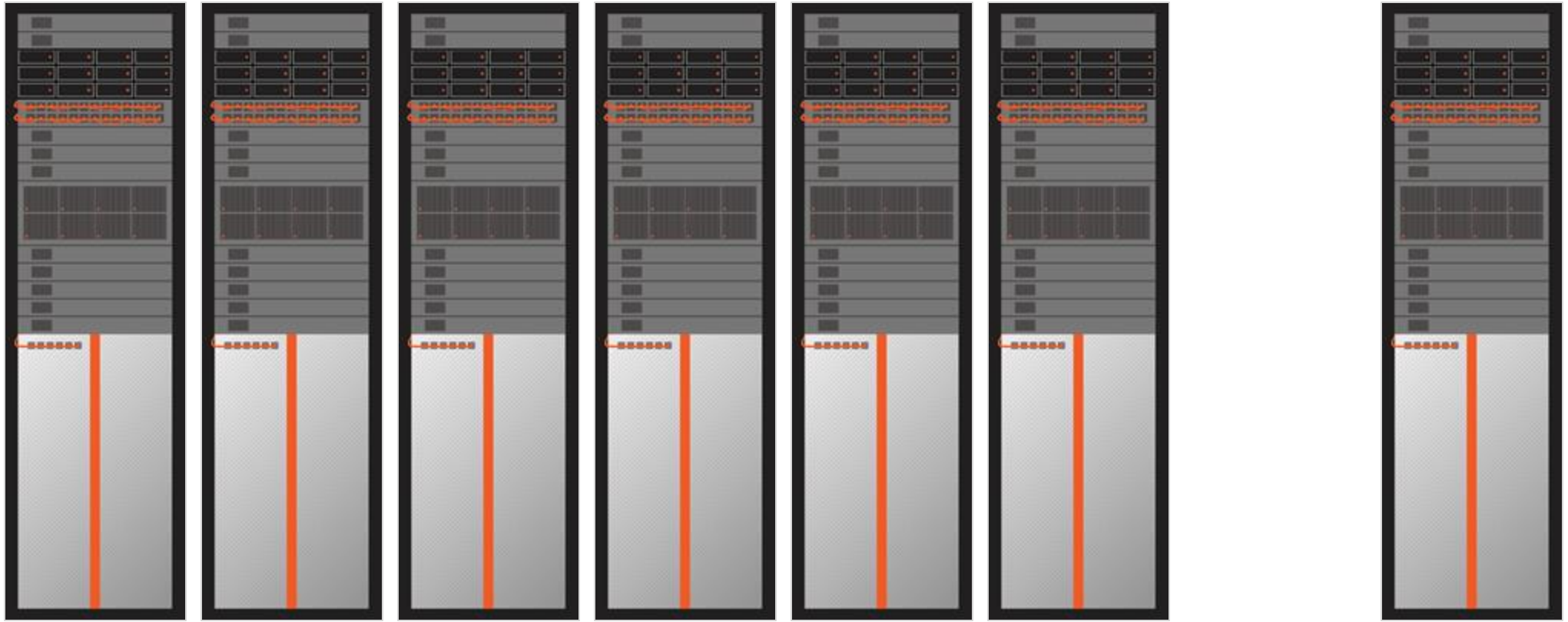
Wafer Scale Cluster: Purpose Built for Wafer-Scale



Wafer Scale Cluster: The World's Most Scalable AI Supercomputer

MemoryX	1 terabyte	1 petabyte
SwarmX	1 CS-3	2048 CS-3s
CS-3	125 petaflops	256 exaflops
	1 billion parameters	24 trillion parameters

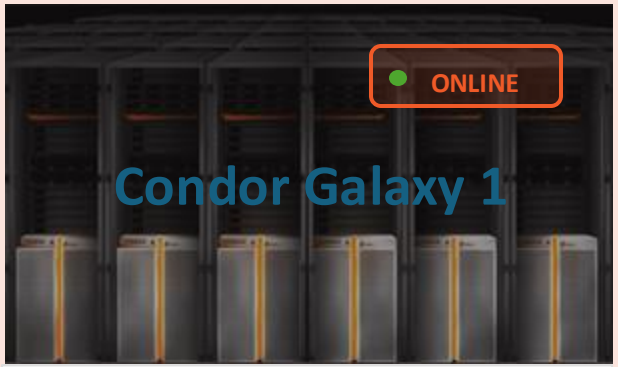
Exa-scale Performance



cerebras

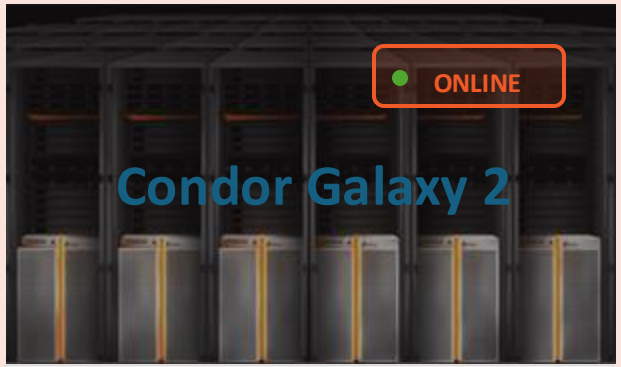
AI Supercomputers
Built & Operated in the United States

Santa Clara, CA



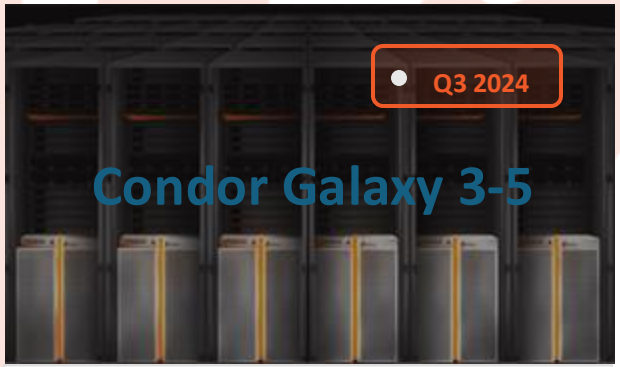
- 4 ExaFLOPs
- 64 x CS-2s
- 82 TB of Memory

Stockton, CA



- 4 ExaFLOPs
- 64 x CS-2s
- 82 TB of Memory

Dallas, TX



- 24 ExaFLOPs
- 192 x CS-3s
- 324 TB of Memory

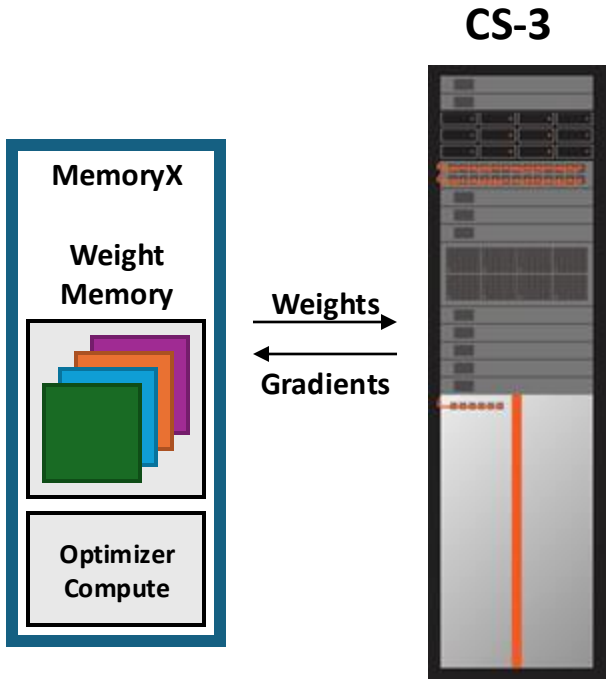
Cluster Components

MemoryX External Memory

Virtually Unlimited Model Weight Capacity

Model capacity not limited by device

- Weights streamed onto wafer to compute layer
- Weights trigger compute using HW dataflow
- Weights are never stored on wafer



Decoupling weight optimizer compute

Memory hierarchy capable of massive models on single device

- Weight update occurs in MemoryX

Enabling Multi-Trillion Parameter Models

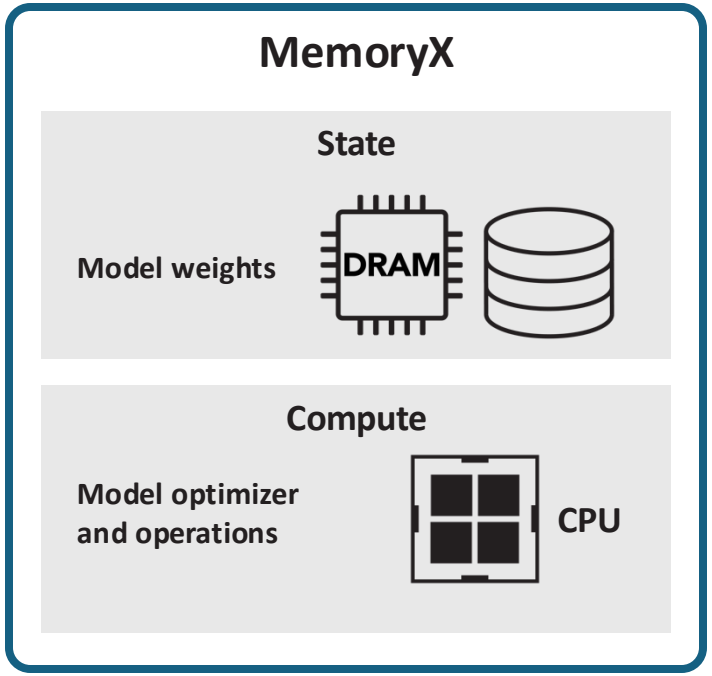
Most Scalable and Efficient Model Memory

Efficient hybrid state store

- **Weights stored in DDR5 and Flash**
- **Perf and power/cost efficiency**

Flexible compute

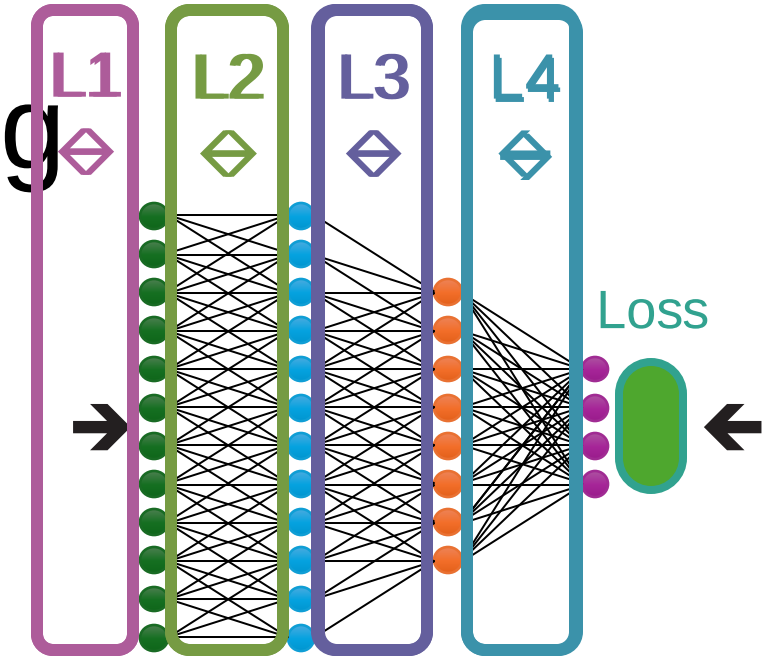
- **Optimizer and other ops run on CPU**
- **General purpose and flexible**
- **Support for all common ML ops**

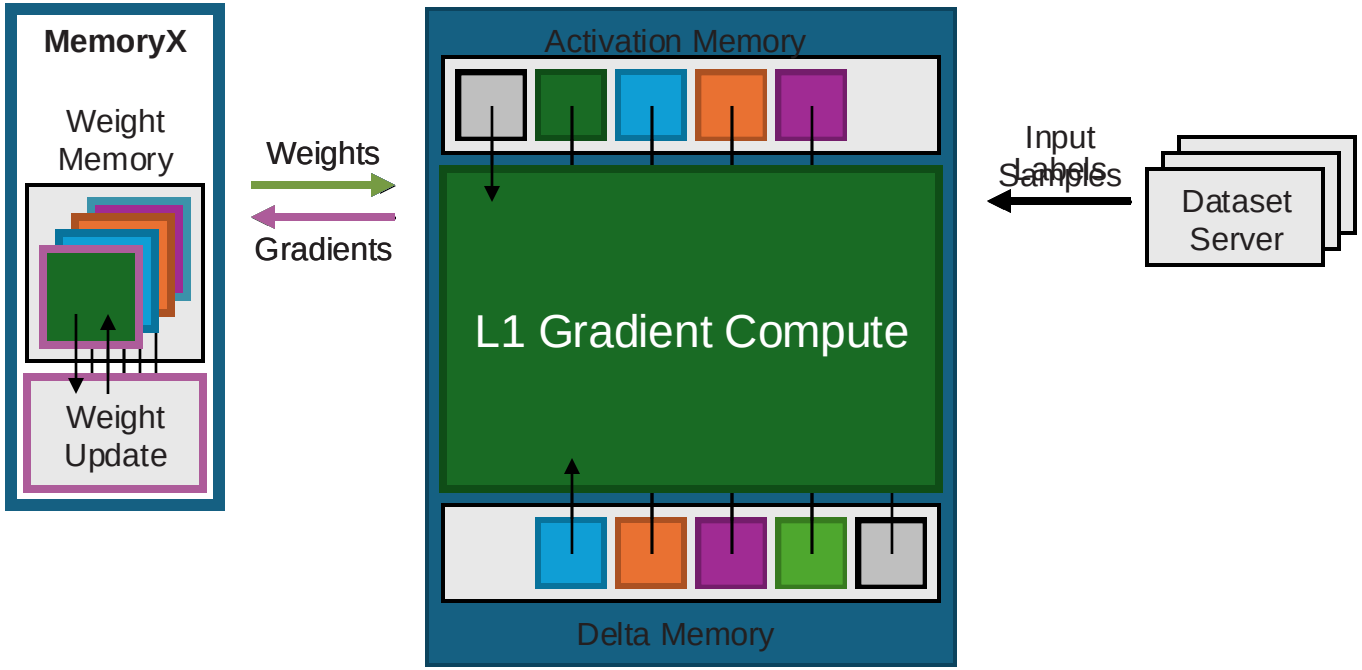


Cluster Options

	CS-2	CS-3
DRAM Memory	12 TB DDR4 240B params	**36 TB DDR5** **720B params**
Flash Memory		**1.2 PB** **24T params**
CPU Perf	1x	**2x**

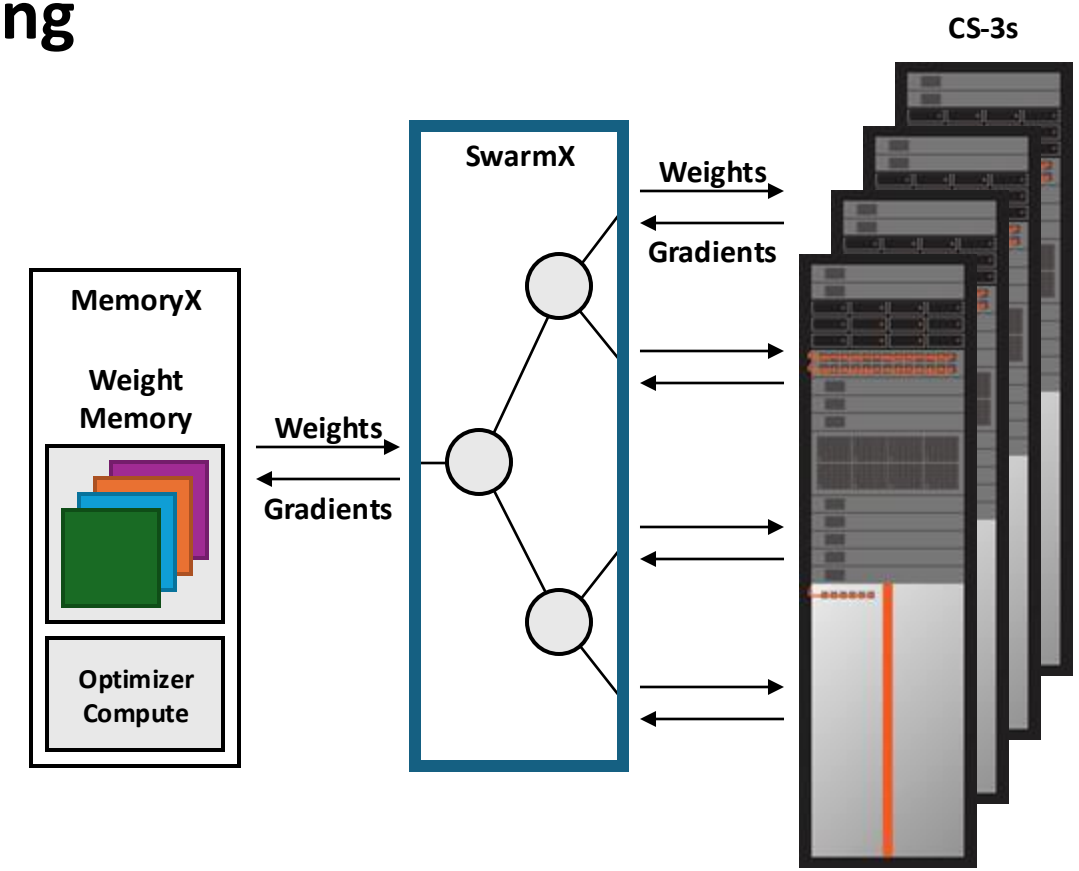
Weight Streaming In Action





SwarmX Fabric

Purpose Built Interconnect for Simple Scaling



Data-parallel only training across CS-3s

- Weights are broadcast to all CS-3s
- Gradients are reduced on way back

Multi-system scaling with the same execution model as single system

- Same system architecture

Scaling to cluster compute while operating like a single device

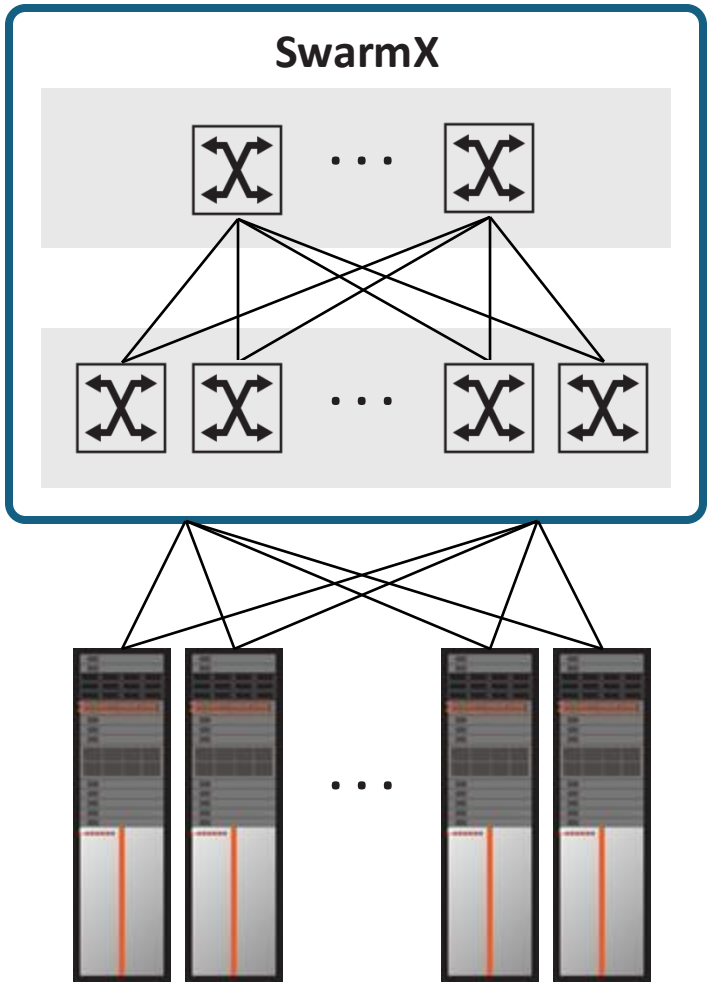
- Same software user interface

Scaling to 256 exaFLOPS

Purpose Built Scalable Network for AI Training

Scalable spine-leaf topology

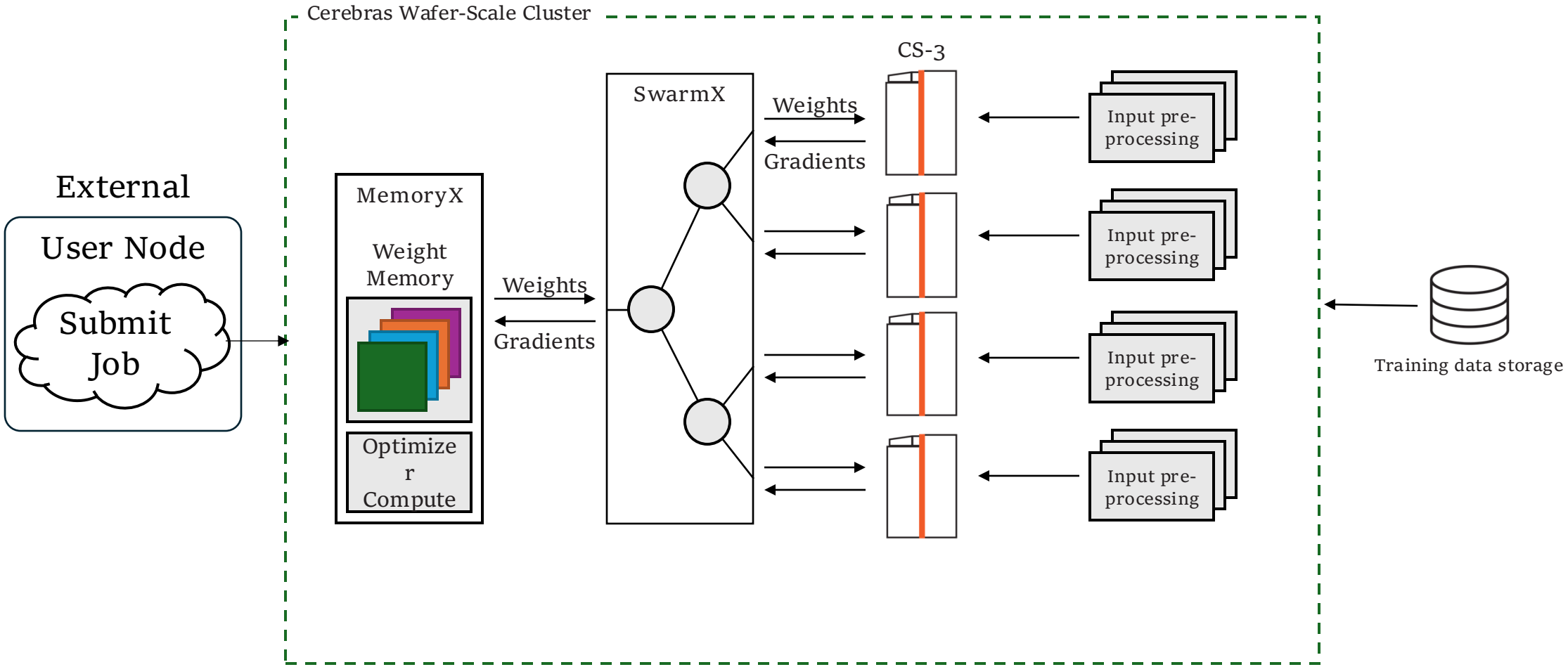
- **Standard-based 400/800G Ethernet**
- **Performance and cost effective**
- **RDMA for low overhead and latency**



Cluster Options

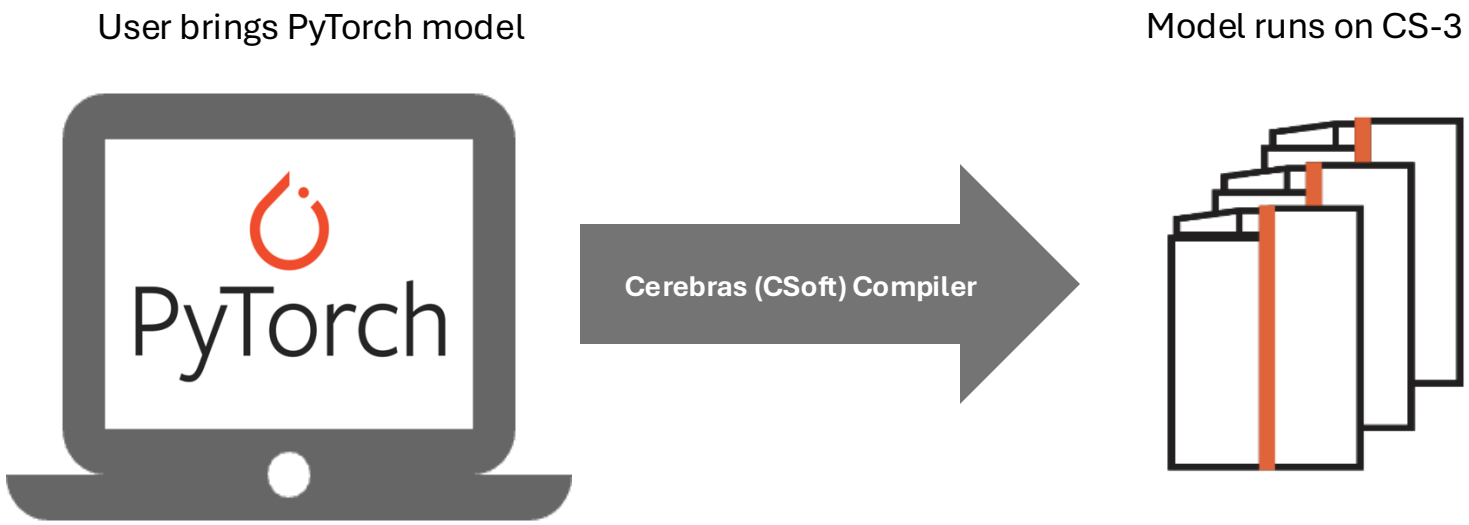
	CS-2	CS-3
Cluster Size	192 systems	2048 systems
Link Speed	100 Gb/s	400 Gb/s 800 Gb/s
Cluster Bandwidth	1 Pb/s	10 Pb/s

System Architecture: The Cerebras Wafer-Scale Cluster & Appliance Model



Programmability

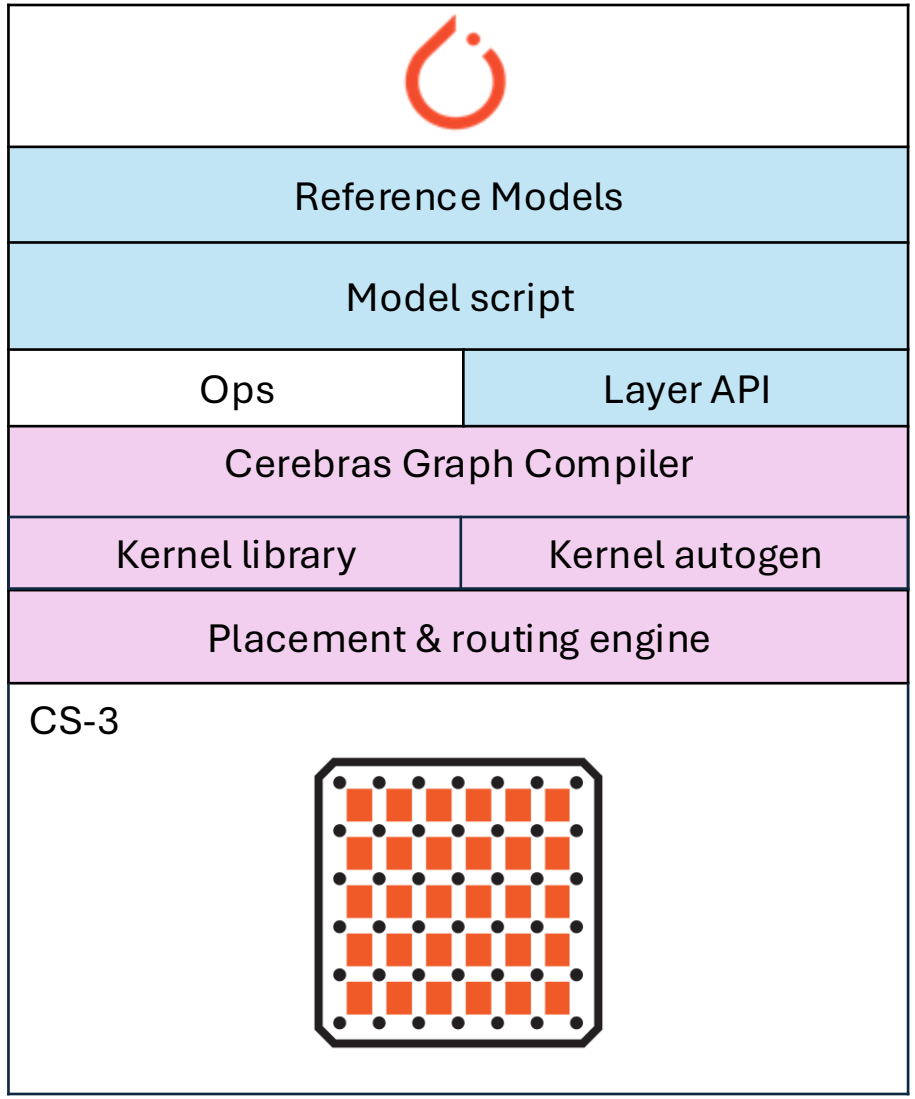
Models Written for GPUs in PyTorch Run Seamlessly on Cerebras Hardware



Cerebras Software Workflow: From Model to Wafer

Integration with PyTorch

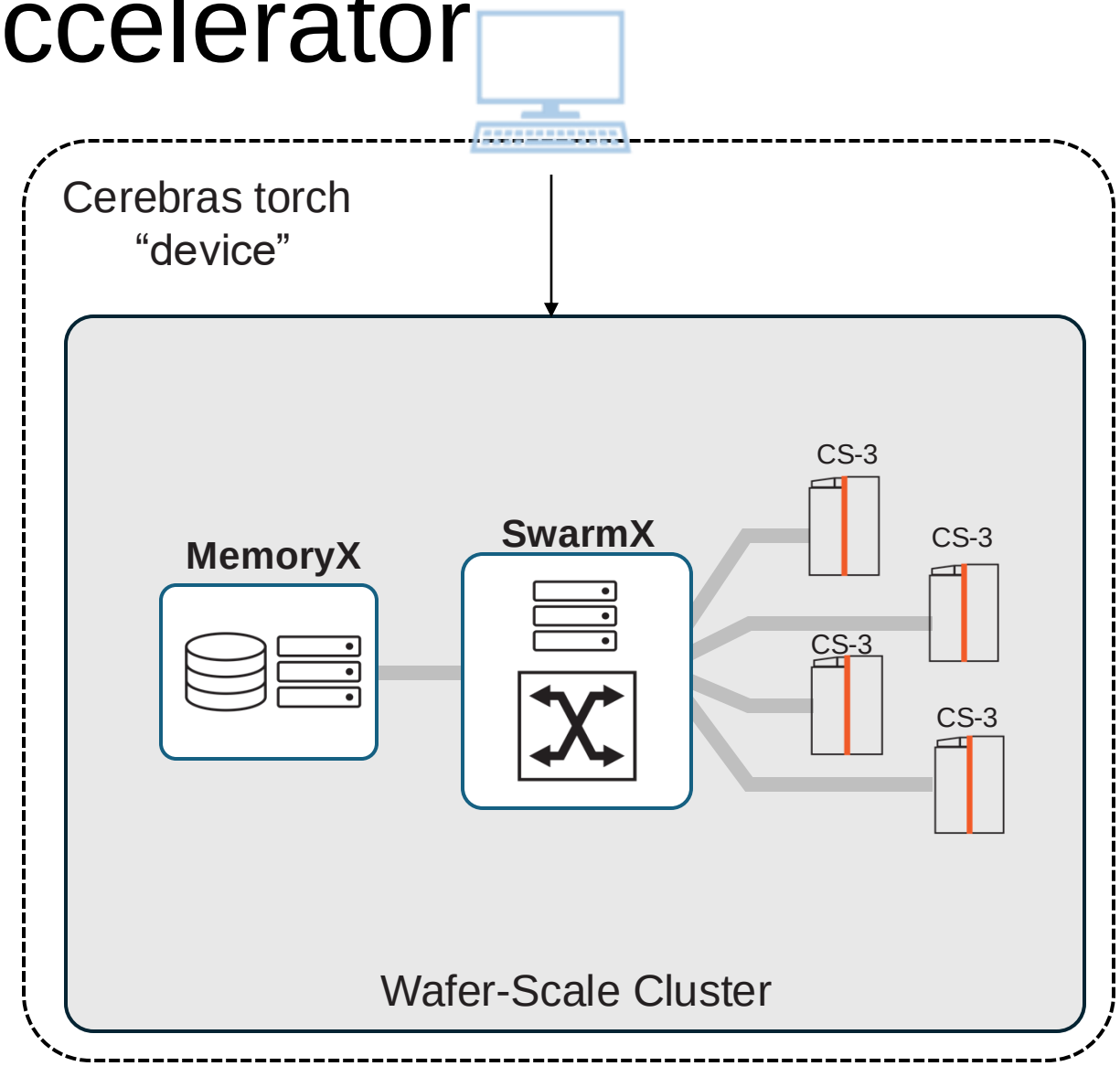
- Models defined in framework + Cerebras API
- Optimally maps from PyTorch to high performance kernels
  - Uses polyhedral code-generation or hand-written kernels
- Compiler using industry standard MLIR framework
  - Cerebras is an active contributor to the MLIR open- source community
- User does not worry about distributed compute or parallelism



The Cluster *is* the ML Accelerator

Disaggregation of memory and compute

- Architect cluster-level memory and compute
- Store model weights externally
- Untangle memory and compute dependency
- Cluster acts a remote accelerator
- All complexity of distributed training, data parallelism, scaling are handled automatically by the Wafer-Scale Cluster
- Exposed as a single device to PyTorch



How to scale from 1B to 70B on Cerebras

gpt3_1b_params.yaml

```
### GPT-3 XL 1.3B

hidden_size: 2048
num_hidden_layers: 24
num_heads: 16
```

llama2_70b_params.yaml

```
### Llama-2 70B

hidden_size: 8192
num_hidden_layers: 80
num_heads: 64
```

Training:

```
python run.py \
--params gpt3_1b_params.yaml \
--num_steps=100 \
--model_dir=model_dir \
```

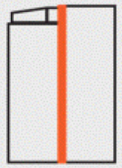
Training:

```
python run.py \
--params llama2_70B_params.yaml \
--num_steps=100 \
--model_dir=model_dir \
```

Scaling from one CS-3 to a cluster is a 1-line change

```
python run.py
--params params.yaml
--num_csx = 1
--model_dir = model_dir
--num_steps = 1000
--mode=train
```

How many nodes?



Resulting in ~~Near~~ Linear Scaling

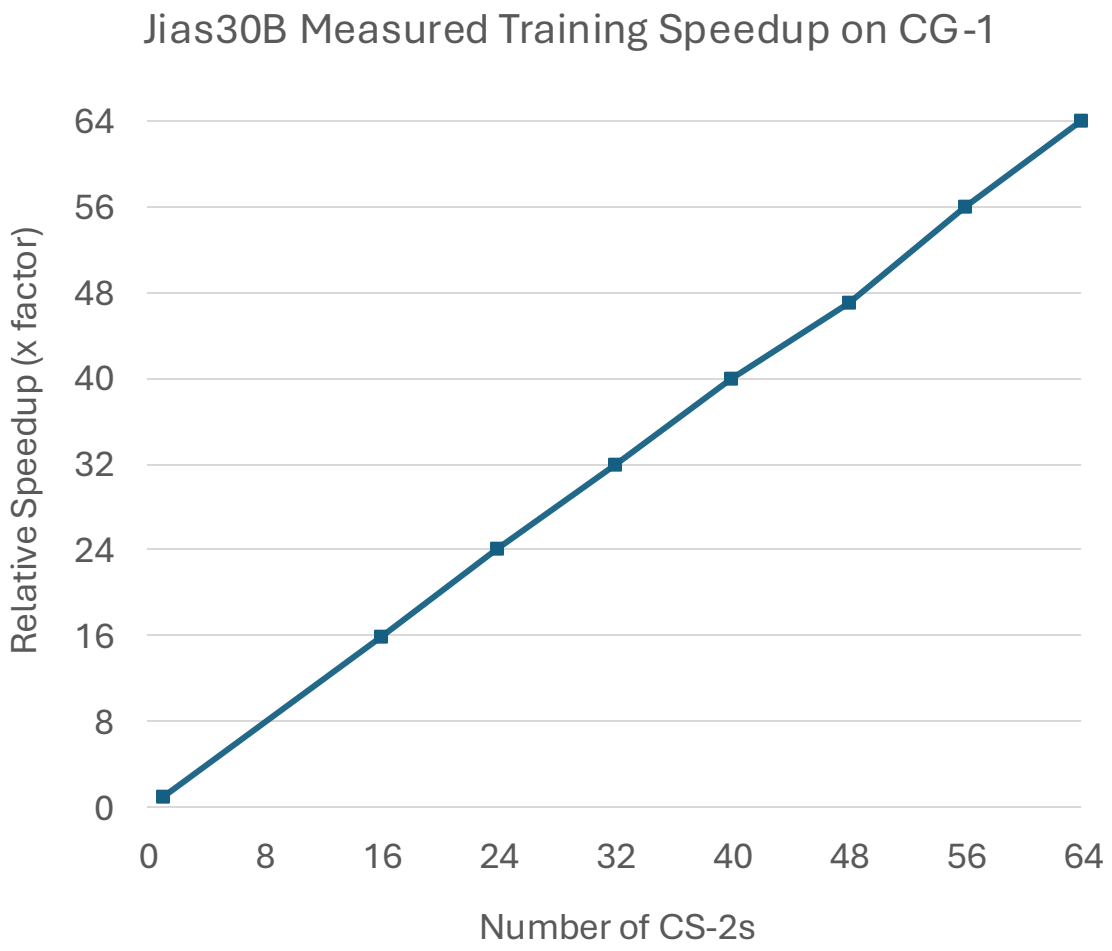
Any Scale While Operating as a Single Device

Real world seamless cluster scaling

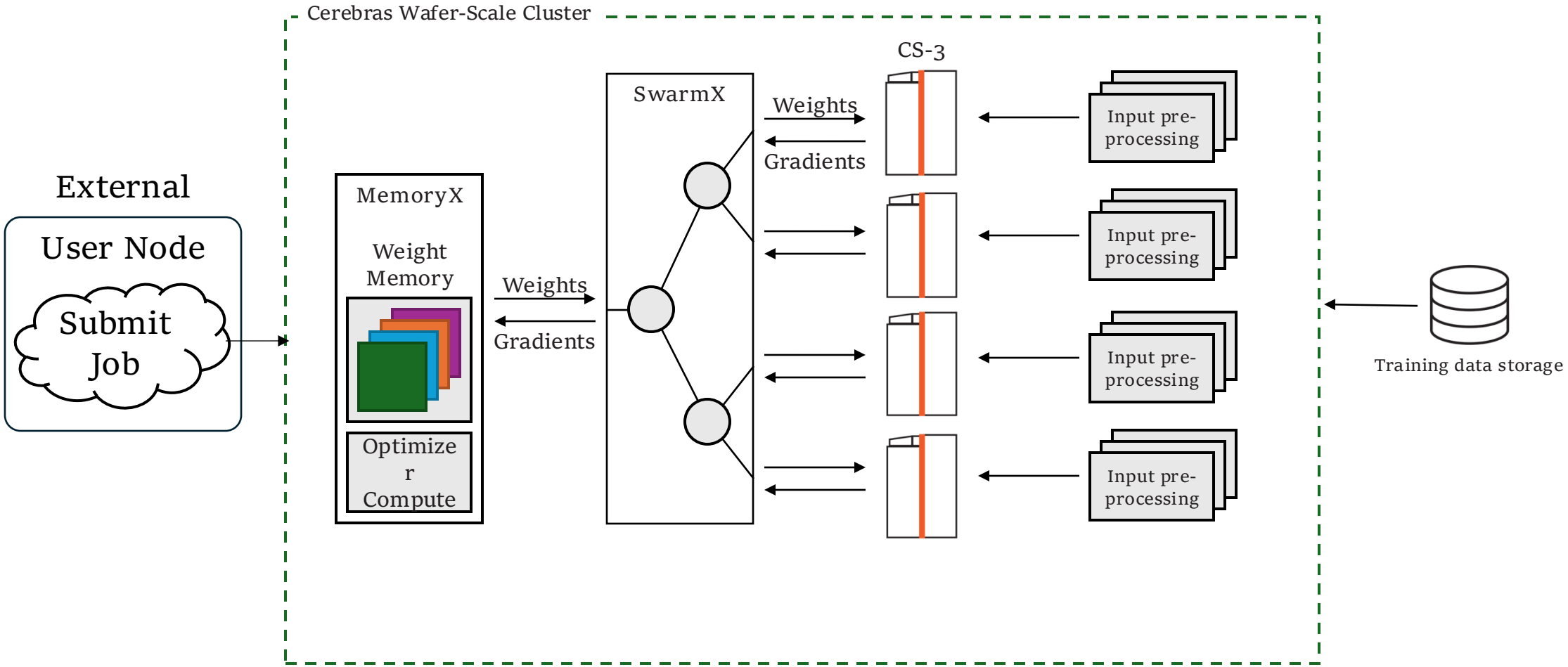
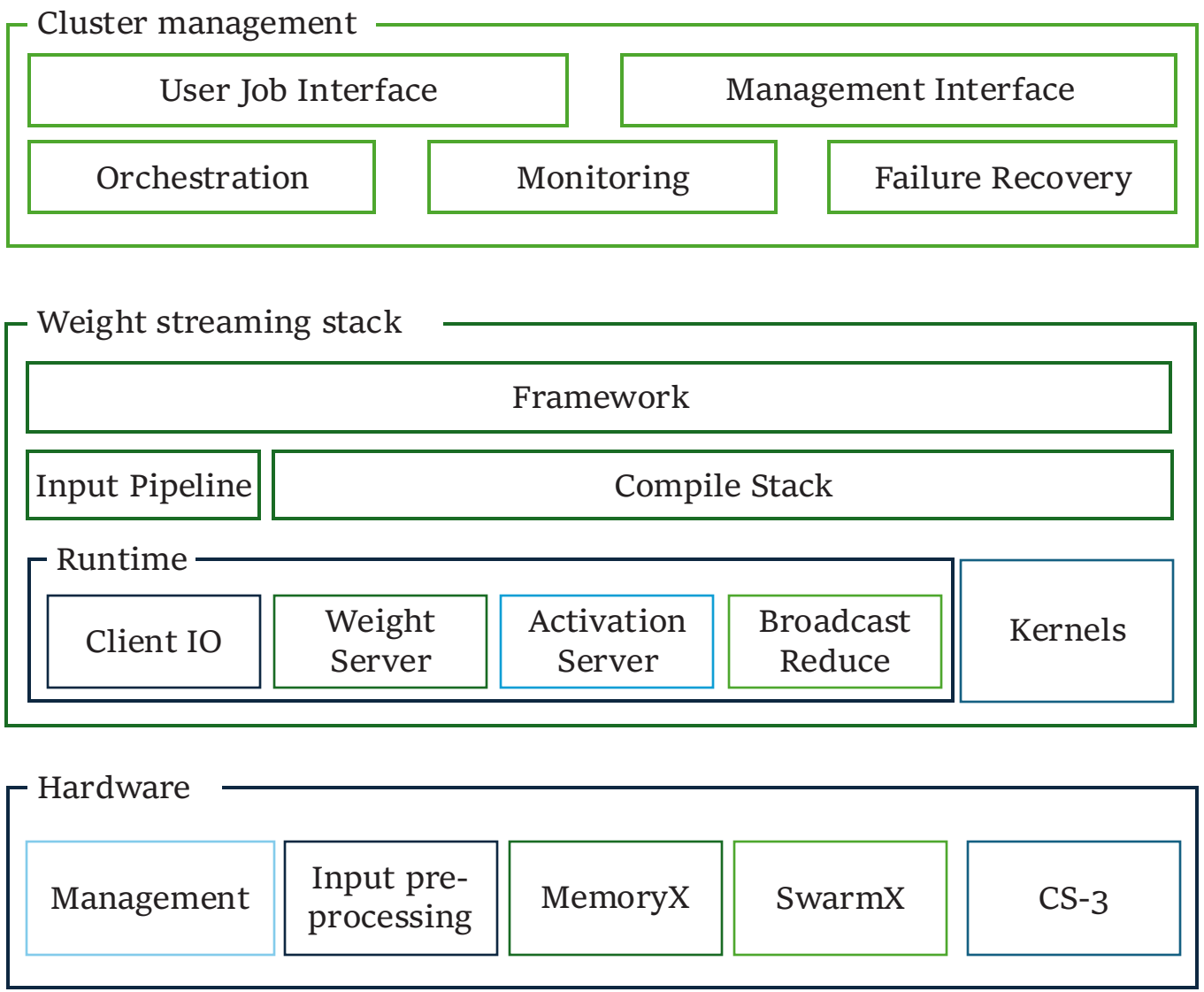
- User: G42
- Model: Jais30B
- Cluster: Condor Galaxy-1
- Experience: “It just worked”
  - No complex distributed software
  - No changes to parallelism model
  - No changes to hyper-parameters

Training SOTA large models everyday

- Unique capability enabled by wafer-scale



System Architecture: The Cerebras Wafer-Scale Cluster & Appliance Model



Software Stack in Wafer-Scale Cluster