Dan Rabinovitsj

VP, Engineering Meta

Resilient AI

Building Fault-Tolerant AI Systems

Artificial intelligence (AI) is having quite a moment

Text-to-image generation

Large Language Models (LLMs)

AI-enabled creation tools

Source: Meta for Business. 'Culture Rising: 2023 Trends Report.' 2023.

The Challenge of Scale: Llama 3's Infrastructure

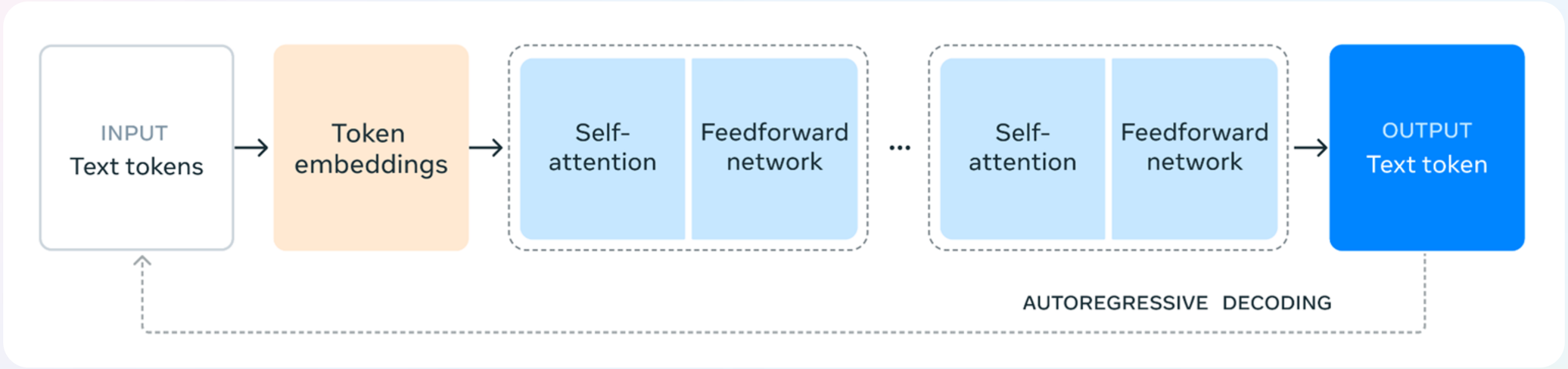
TRAINED AT
UNPRECEDENTED SCALE

>15T tokens

used to train Llama 3.1 405B

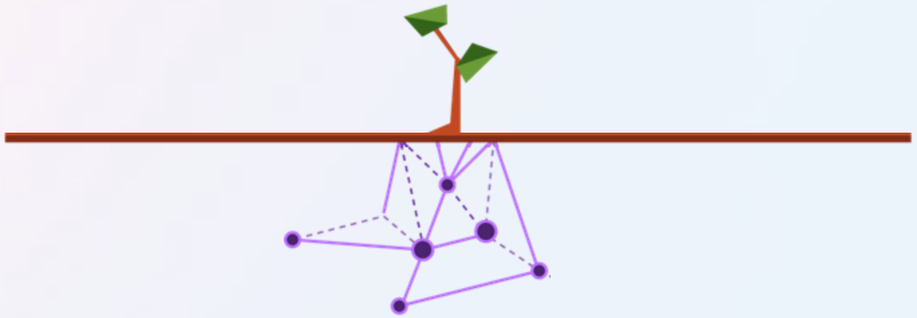
16K H100 GPUs

pushed our model training to
new heights, leveraging a significantly optimized
full training stack



AI jobs at scale: massive change in 2023

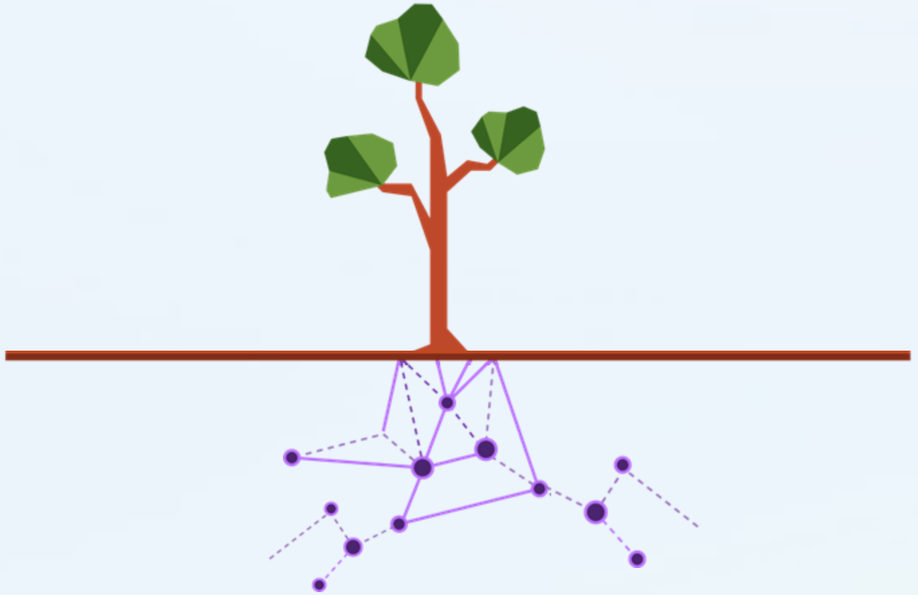
2022



6K clusters

Job size: 128-512 GPUs

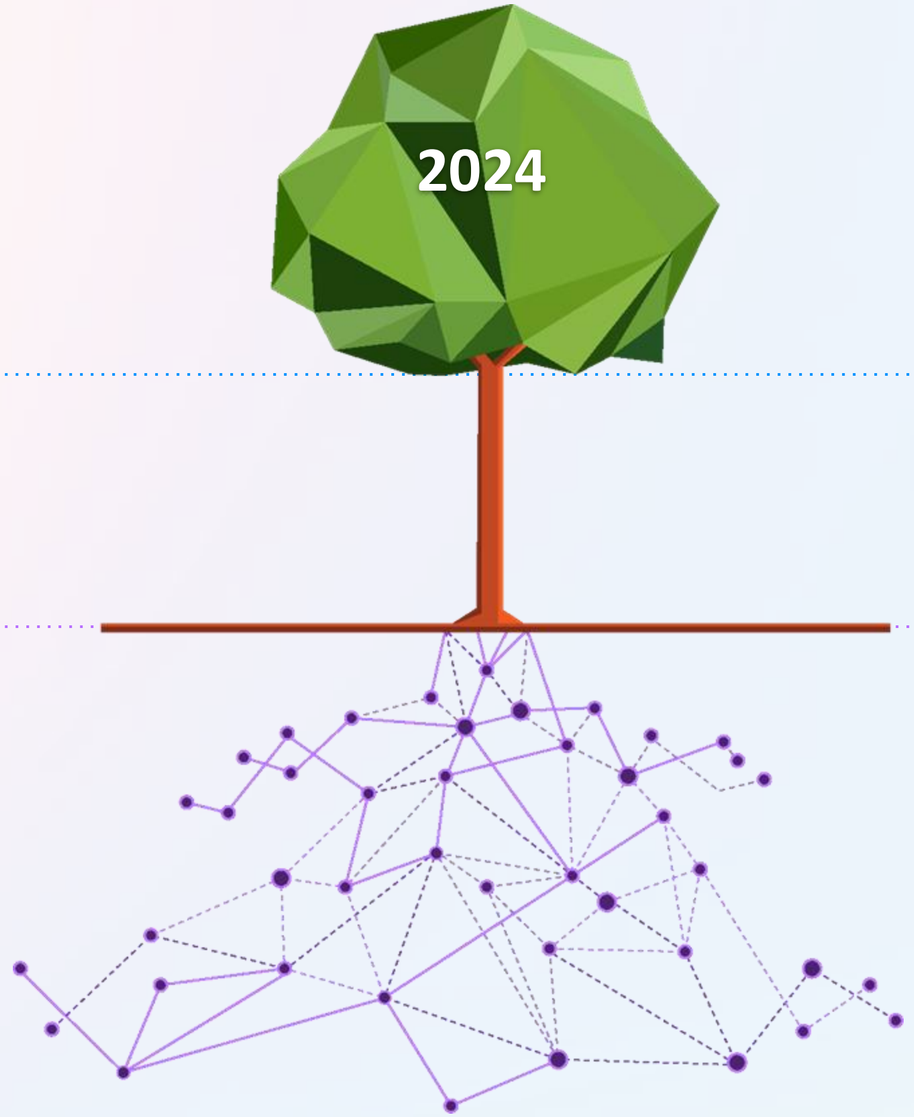
2023



16-24K clusters

Job size: 16K GPUs

AI jobs at scale: Today

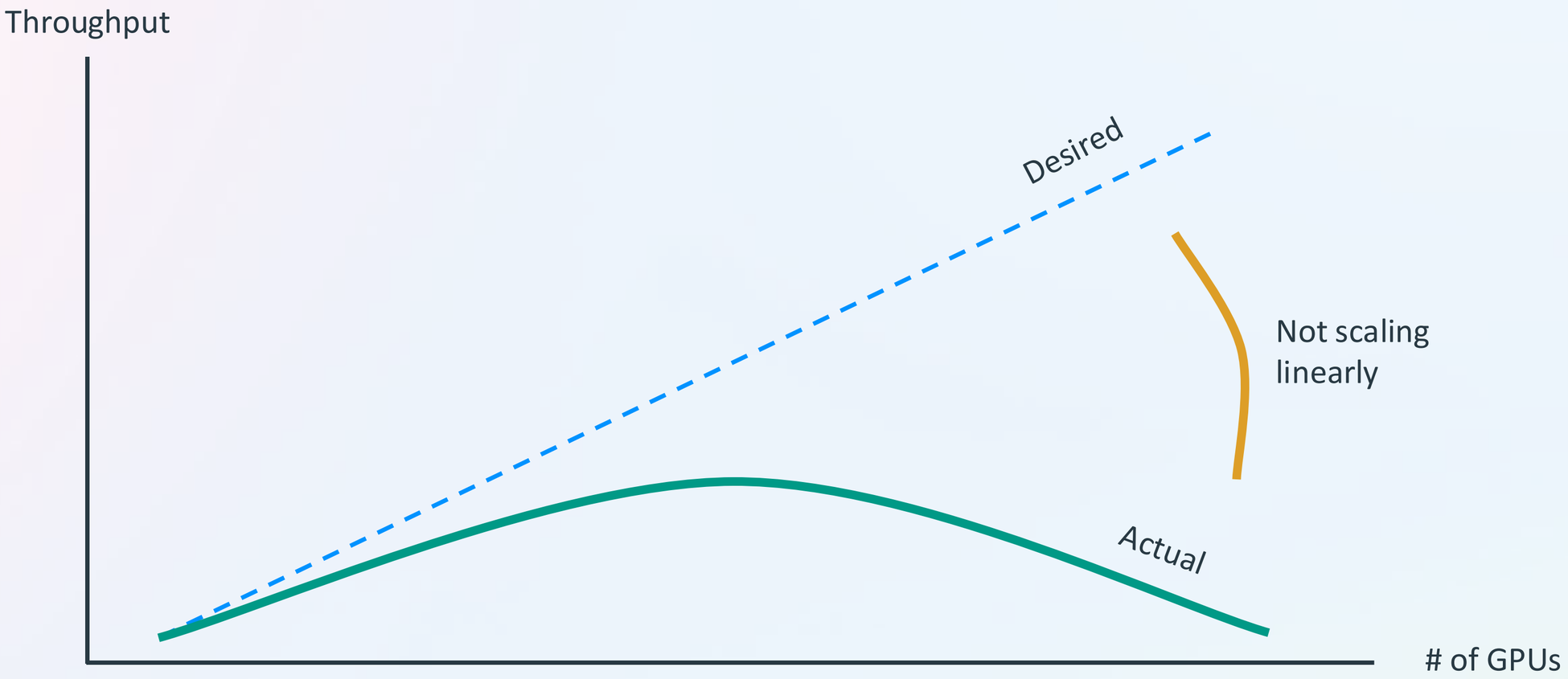


Llama

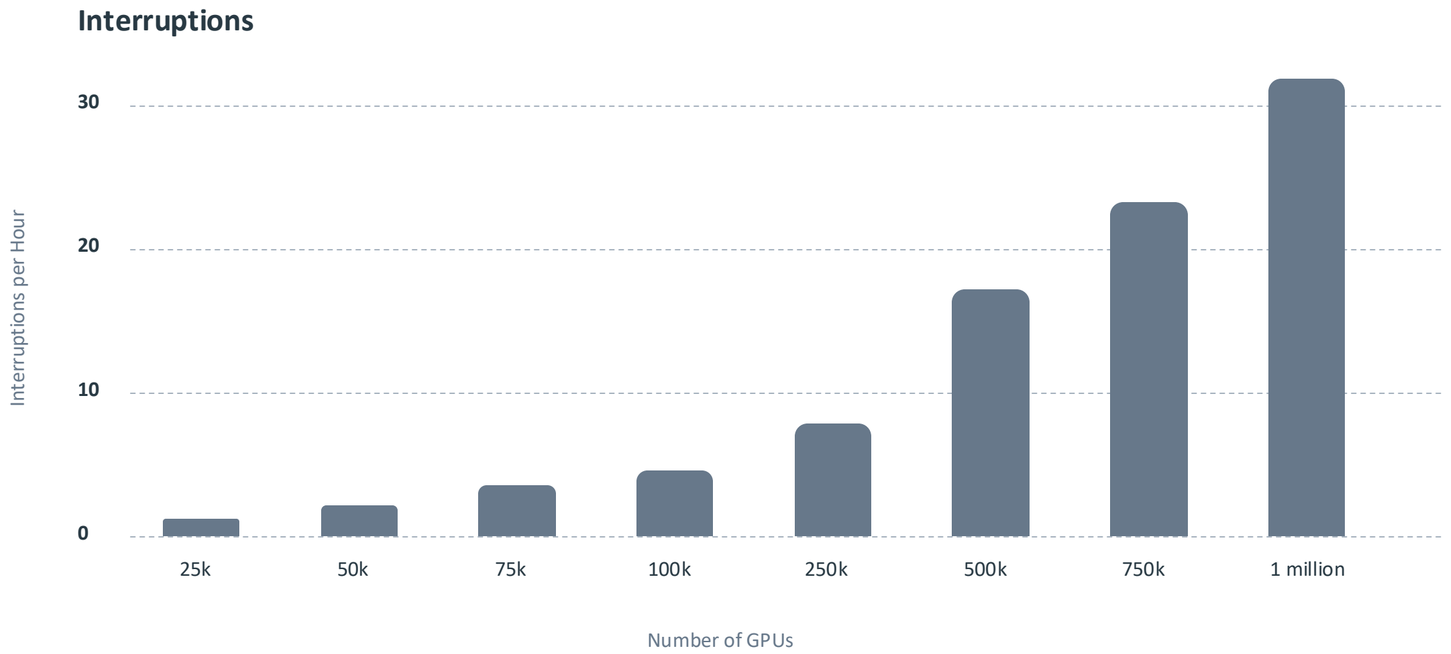
Software Infra

Physical Infrastructure

Training @ Scale is Not Linear!



More GPUs *equals* *More Failures*



Effective Training Time

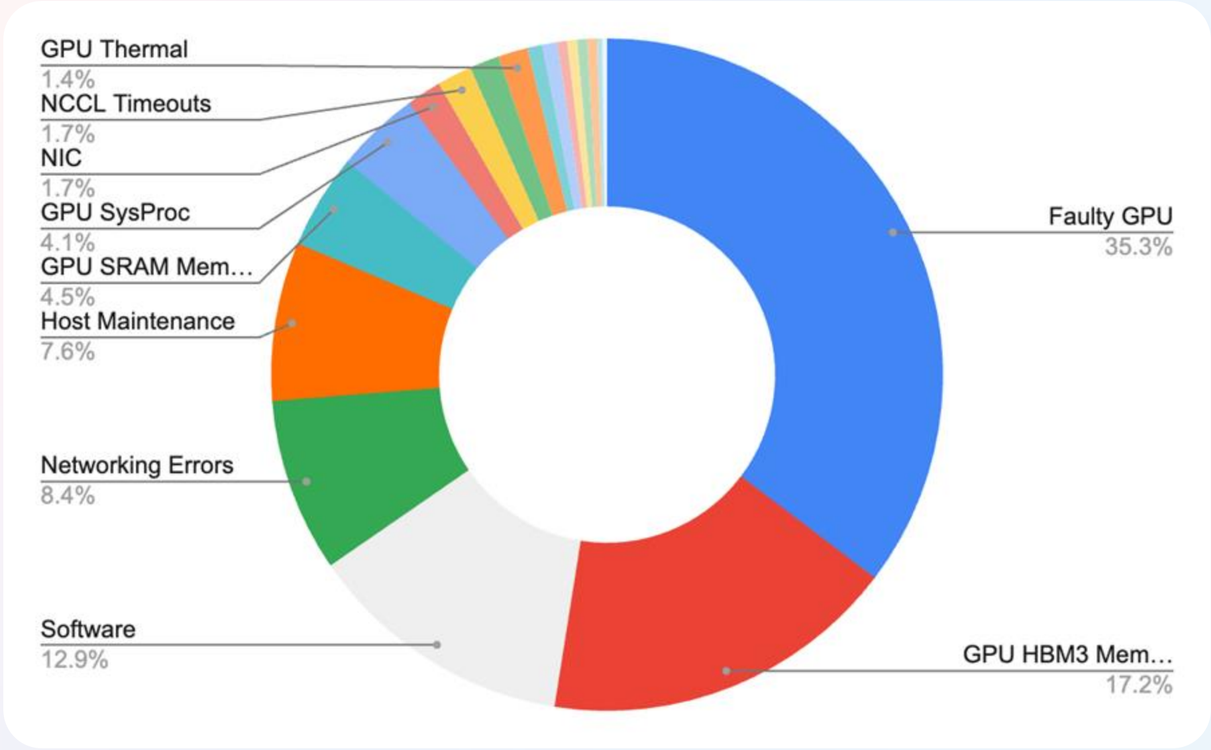
$$1 \text{ minus } \frac{\text{E2E Job Restart Overhead + Lost Training Progress}}{\text{Total Expected Training Time}}$$

AI Infrastructure is **fundamentally different**

	Traditional Infrastructure	**AI Training Infrastructure**
Time to Repair	Days	**Hours**
Degradation Signature	Linear/Graceful	**Non-Linear/Sudden**
Traffic Flows	Mice	**Elephant**
Distributed Comms Orchestrator	N/A	**Critically Necessary**
Workloads	Millions	**One**
Impact of Failures	Minimal	**Severe**

Hardware failures dominate for AI Cluster

We were early adopters and saw more early failures.



The primary causes of these interruptions include:

Hardware Reliability

GPUs are inherently less reliable than CPUs, and a single GPU failure can halt the entire training process.

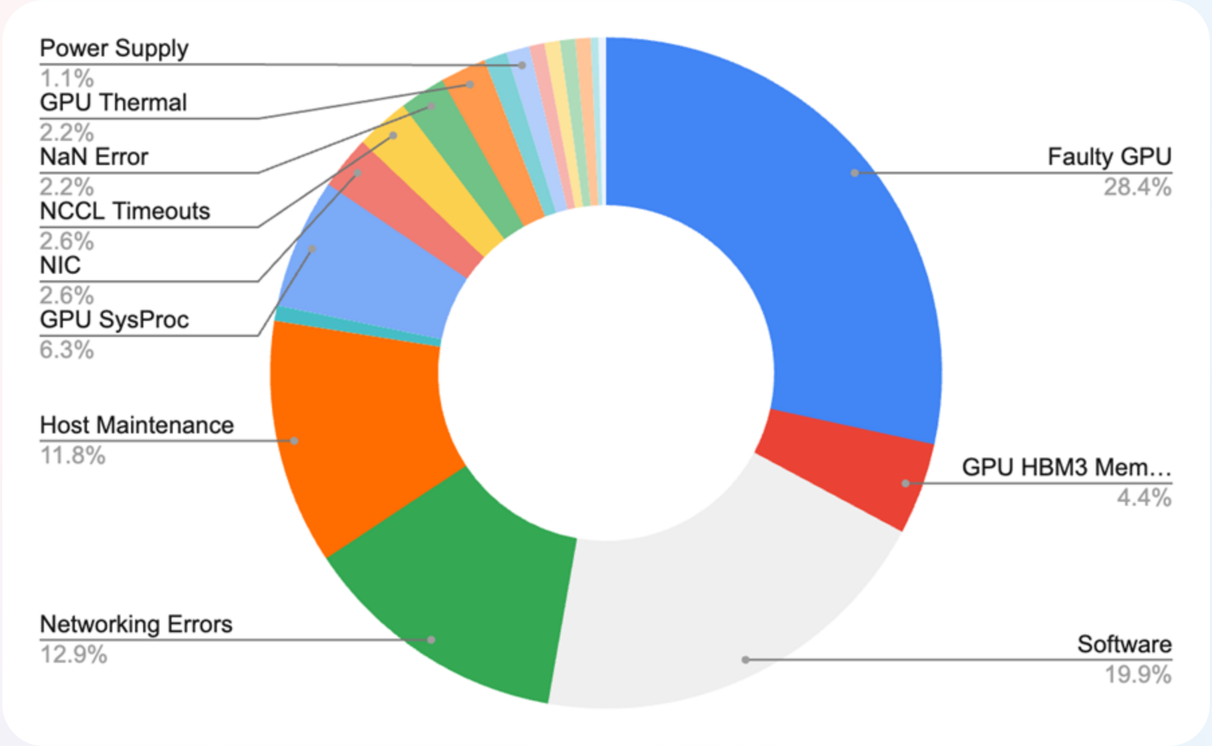
Network Complexity

The network infrastructure is complex, and debugging network issues is time-consuming.

Software Configuration

GPU-related software requires extensive configuration, which can be error-prone.

Hardware failures dominate for AI Cluster



55.4% of Interruptions are **hardware failures**

The primary causes of these interruptions include:

Hardware Reliability

GPUs are inherently less reliable than CPUs, and a single GPU failure can halt the entire training process.

Network Complexity

The network infrastructure is complex, and debugging network issues is time-consuming.

Software Configuration

GPU-related software requires extensive configuration, which can be error-prone.

How do we make this better?

Avoid

Detect

Tolerate

How do we make this better?

Avoid

Detect

Tolerate

Making Rack scale Designs More Reliable

Top of Rack Switch

Rack Management Controller

Accelerator Tray

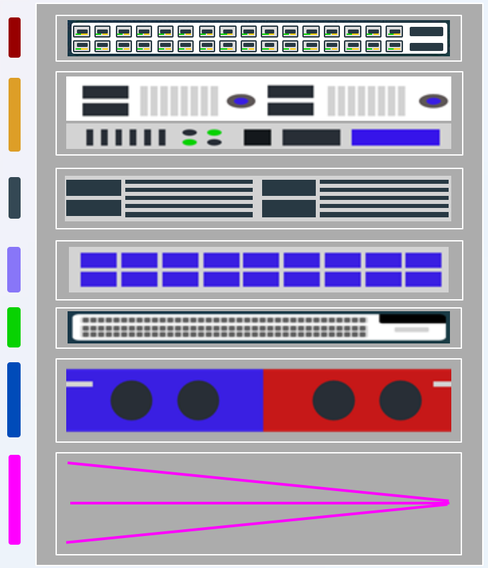
Scale-[Up | Out] Switch Tray

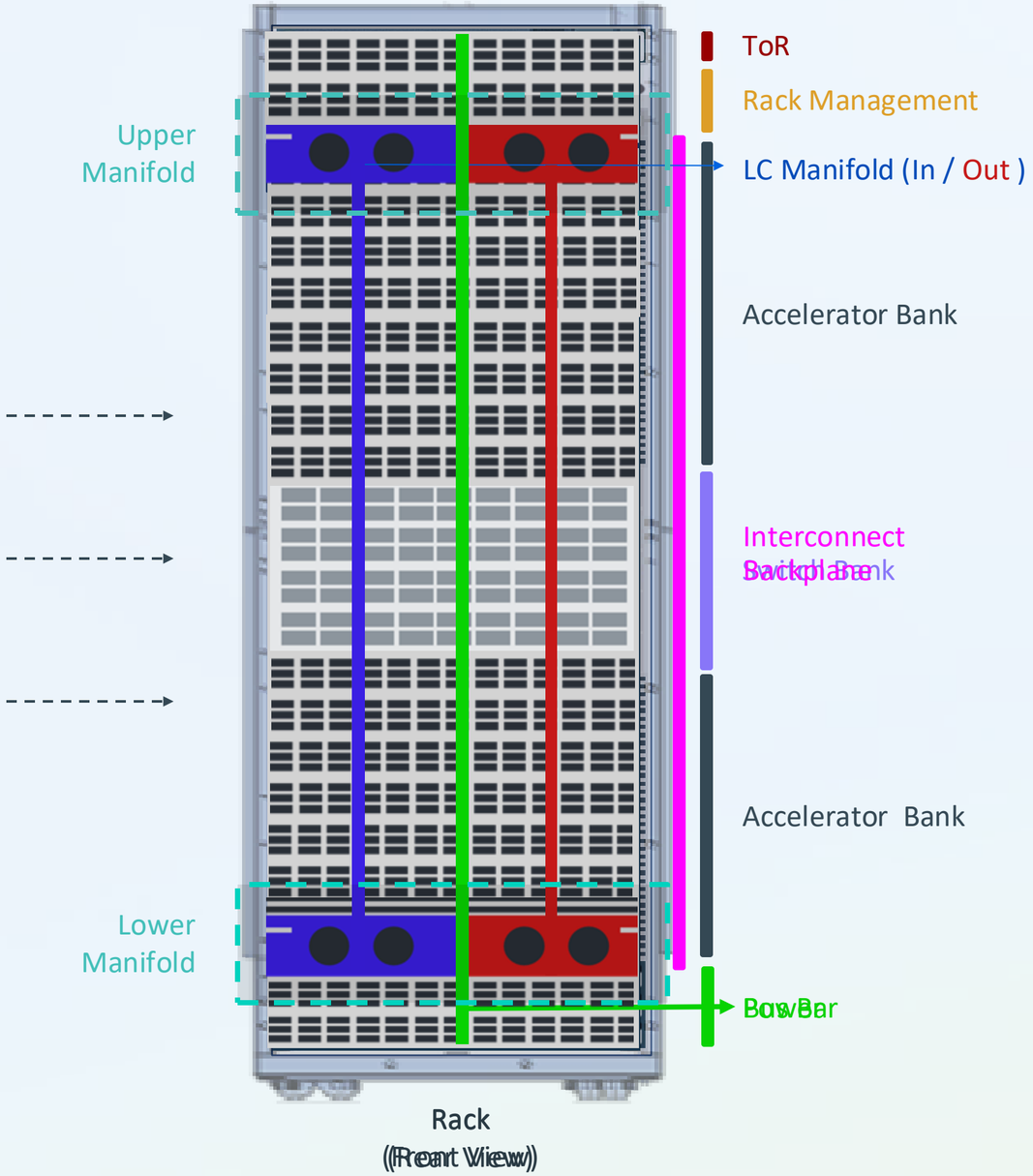
Power and Hold-Up Energy Shelves

Liquid Cooling Manifolds

Interconnect Backplane

Rack Element
“Lego™ Blocks”

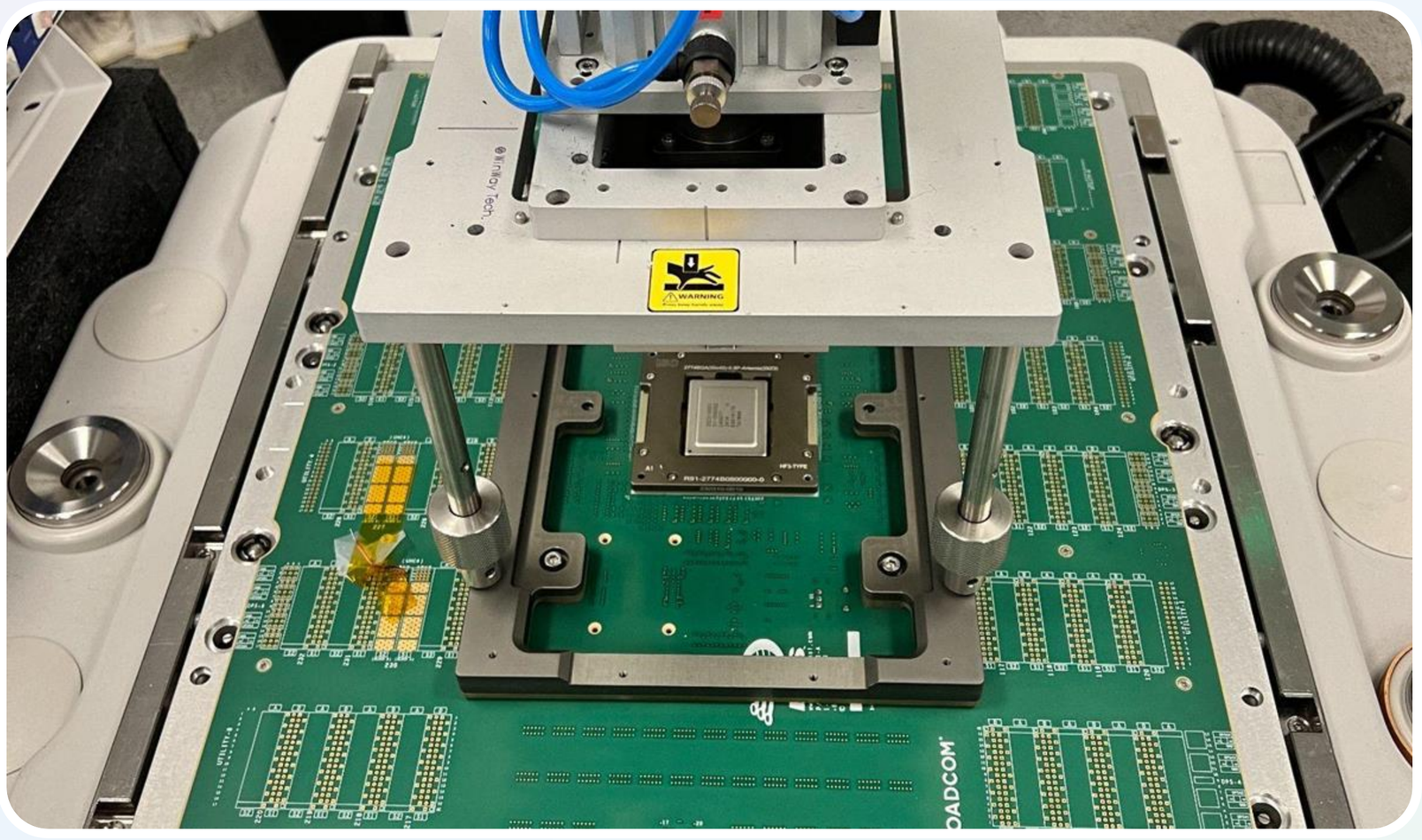




Rack
(Rear View)

Silicon

Test Coverage



L6-10 Board

Test Coverage

L11 Rack

Test Coverage

Cluster Level

Test Coverage

How do we make this better?

Avoid

Detect

Tolerate

“Pre-flight” Checks

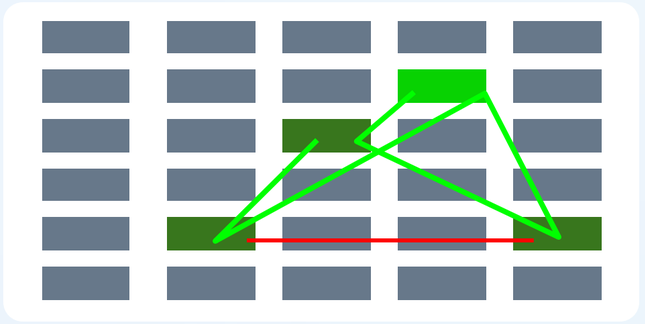
Mast forms Gang of Hosts

- Hosts in AI Zone selected

Individual Hosts are Checked

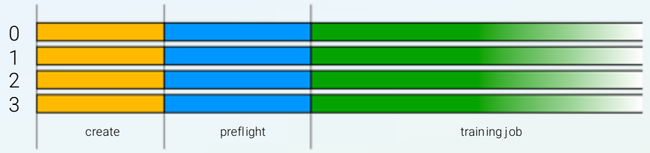
- Are the right number of GPUs visible?
- Any devices inline remediations to perform (e.g reset device)?
- Replace any faulty devices & mark them for maintenance

Distributed Communication Check

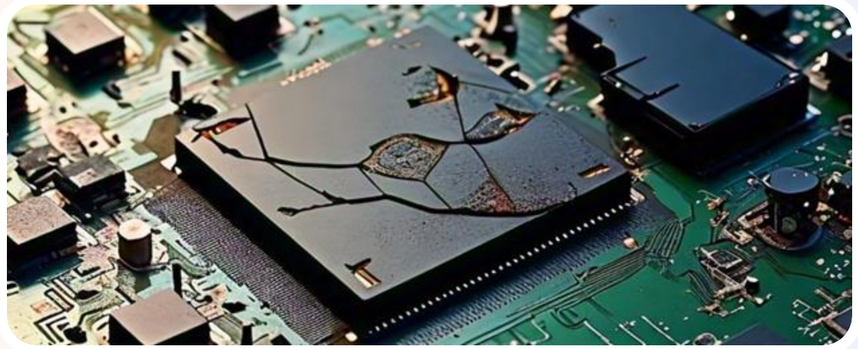


- Run basic communication test
- Run simple (known good) training test

Pre-flight checks allow us to detect system issues rapidly



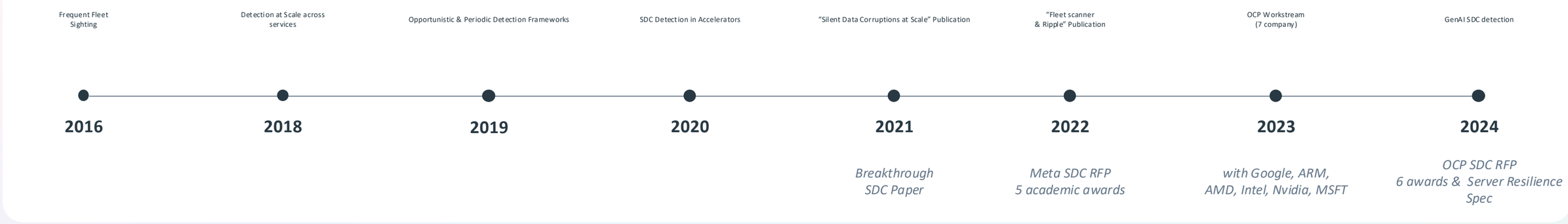
Silent Data Corruption



$$(1.1)^{53} = 0$$

"Defects in silicon leading to silent miscomputations impacting services at scale"

Detection



A journey towards industry leadership and standardization

SDCs at scale - https://arxiv.org/abs/2102.11245
Detecting SDCs in the wild - https://arxiv.org/abs/2203.08989
Meta SDC RFP - https://research.facebook.com/research-awards/2022-silent-data-corruptions-at-scale-request-for-proposals
OCP RFP - https://www.opencompute.org/blog/ocps-server-resilience-initiative-sdc-academic-research-awards-announced

Silent Data Corruption in AI workloads

Impact **Detection**

Training

SDCs

Gradient Explosion
Corruption Propagation

- Algorithmic Fault Tolerance in low-precision GEMMs [IEEE Big Data 2022]
- NaN Detection in PyTorch GitHub
- Periodic and Opportunistic Cluster SDC detection [SELSE 2022]

Inference

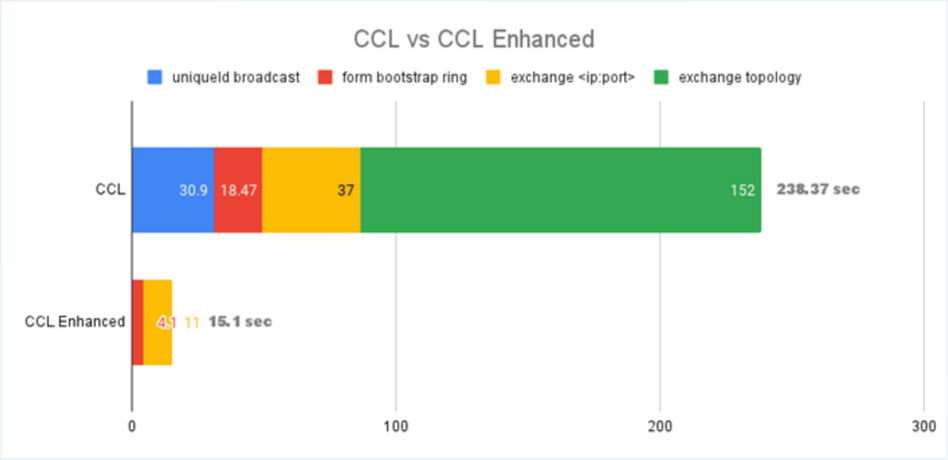
SDCs

Faulty Recommendations
(with impact to privacy, security, policy)

- Parameter Vulnerability Factor driven protection across layers [Meta PVF]
- Dr. DNA – SDC signature (DNA!) distribution based early detection [ASPLOS 2024]

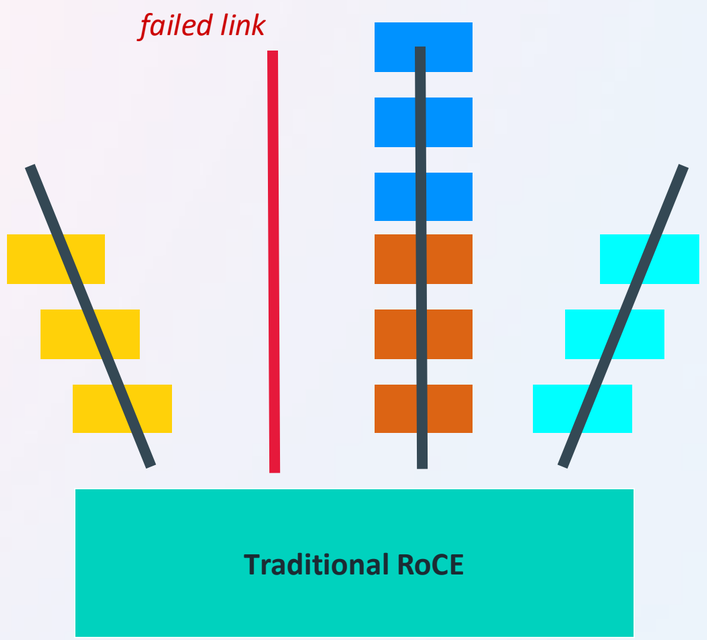
How do we make this better?

Fast Restart

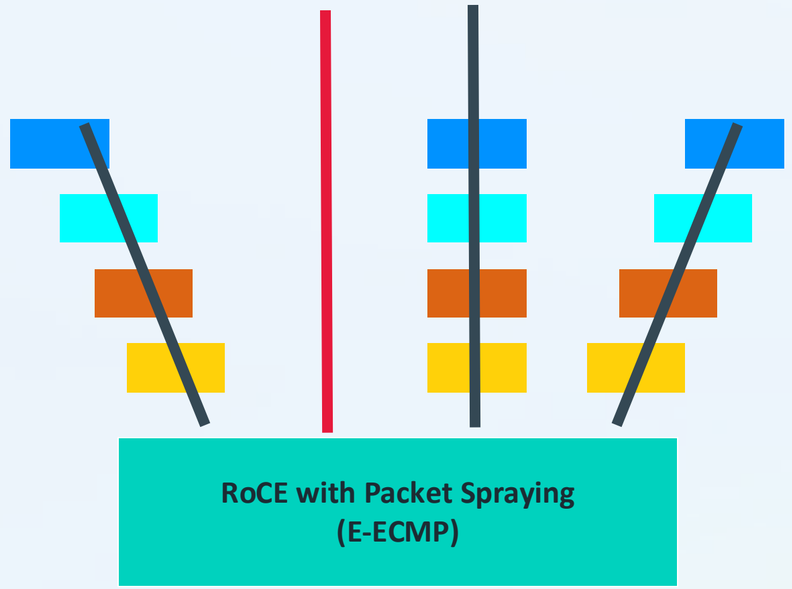


Reducing setup time increases uptime

100% increase
in completion time



33% increase
in completion time



Failure Resilient Transport Key to Fault Tolerant Systems

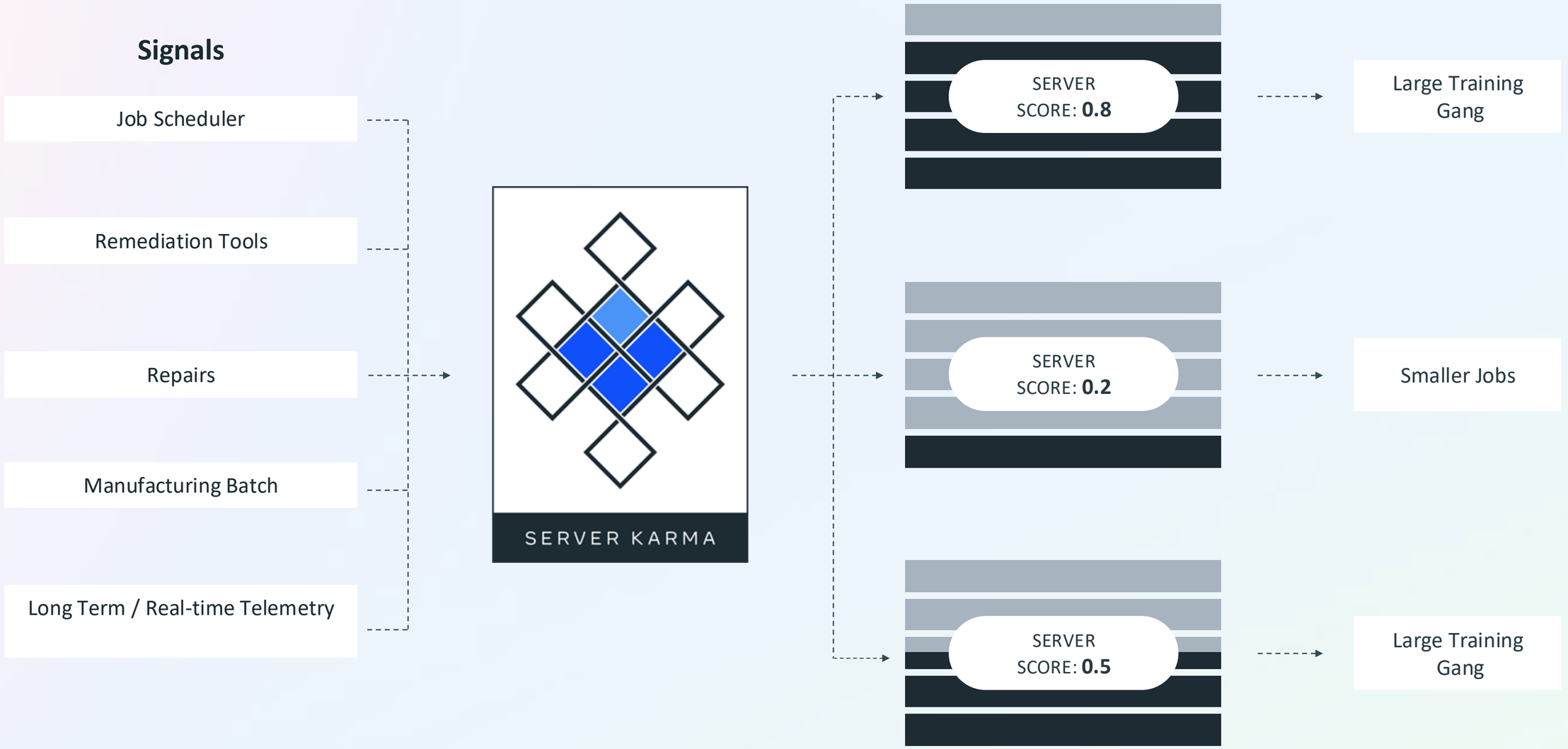
What’s Next?

Hardware Failure Remediation and Recovery

Leveraging AI/ML for System Reliability

Leveraging AI/ML for System Reliability

Server Karma



Better Access to Consistent Debug and Diagnostic Tools

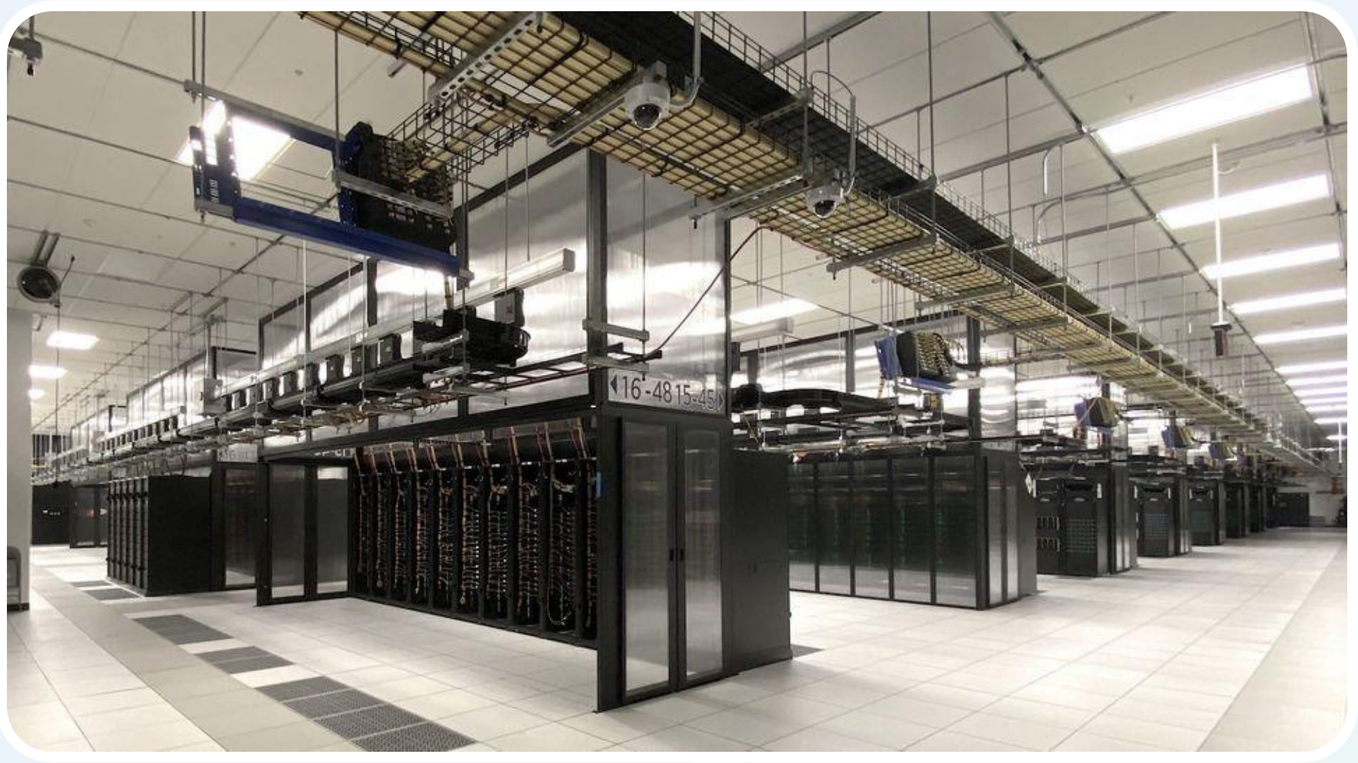
Data Dump Formats

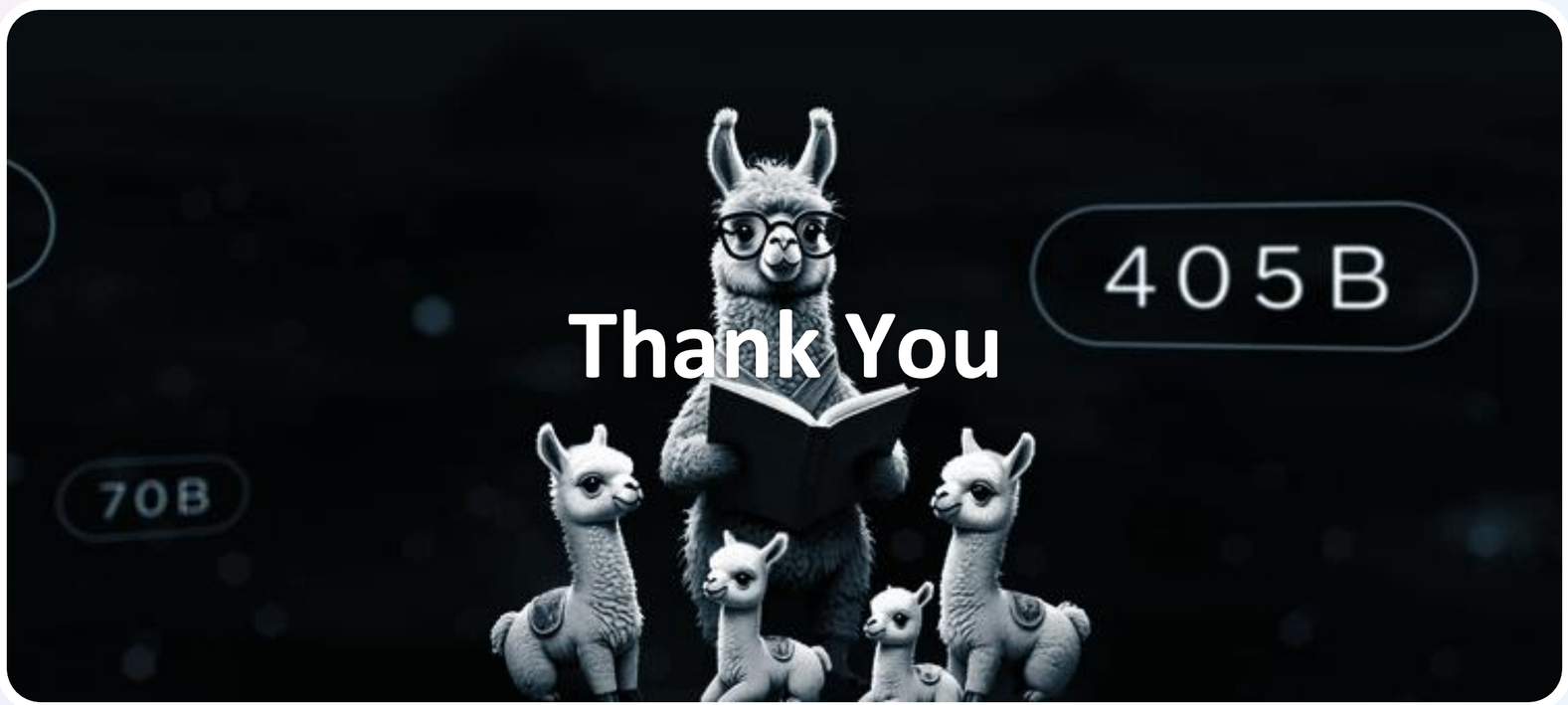
Diag Interfaces

Silent Error Detection

Counters/ Tools

Building Systems with GRIT!





Resilient AI: Building Fault-Tolerant AI Systems

Dan Rabinovitsj


Meta