THE GUTS OF LARGE LANGUAGE MODEL CHECKPOINTING

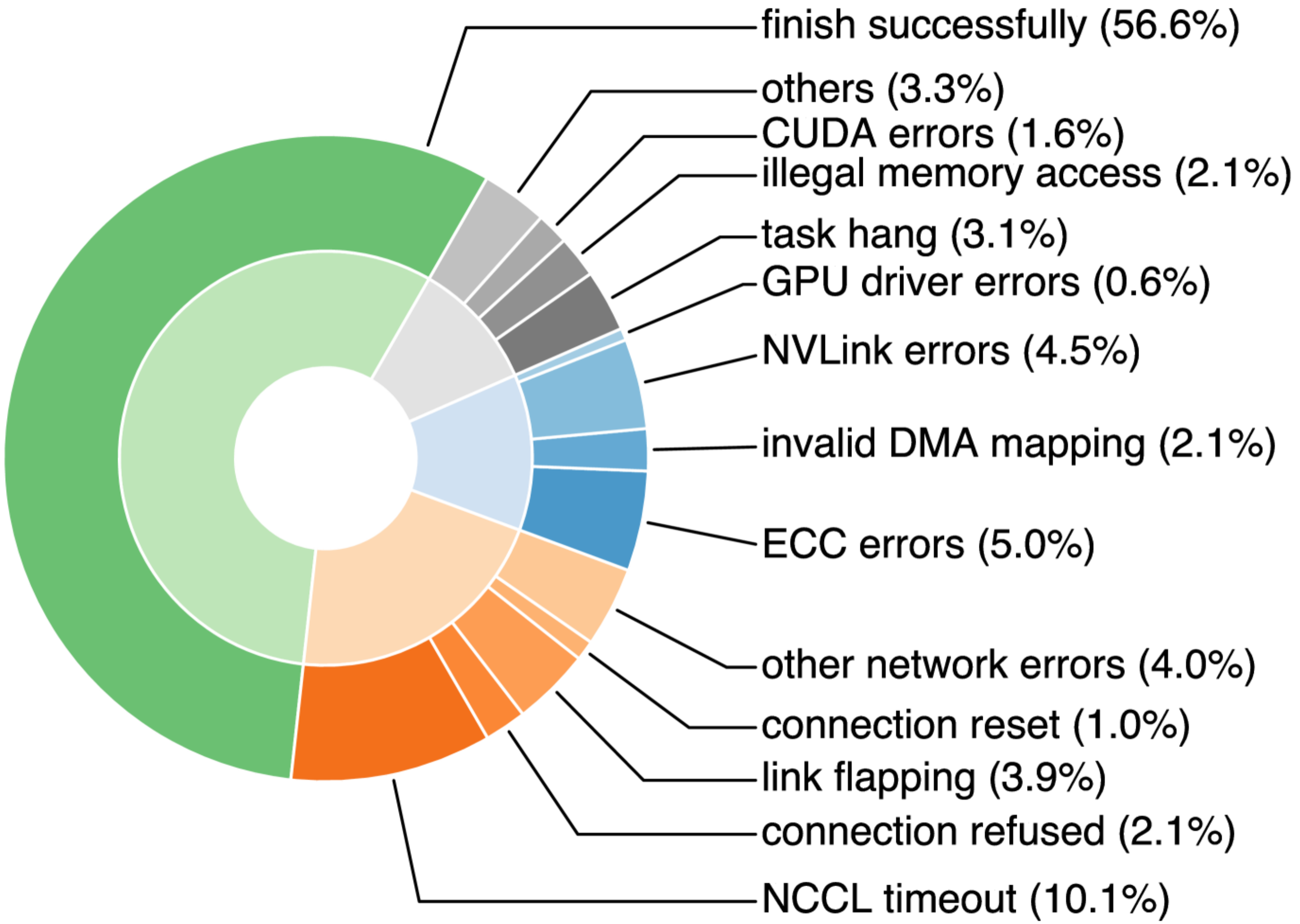
G. Lockwood, Ph.D – Microsoft
S. Kartik, Ph.D – VAST Data

SC24 Atlanta, GA | hpc creates.

THE IMPORTANCE OF LLM CHECKPOINTS

- Significance of LLM Checkpoints
  - Essential for managing long training durations and resource consumption
  - Prevents loss of progress due to inevitable hardware and software failures
- Training a 200B Parameter Model
  - Takes over a month with 1 trillion tokens and 1000 H100 GPUs
- Failure Rates of Long Runs
  - Alibaba's statistics show only 56% success rate
  - Hardware and software issues lead to frequent failures in large-scale environments

STATISTICS ON TRAINING SUCCESS RATES:



CHECKPOINTING: SAVING THE STATE OF TRAINING JOBS

- Importance of Saving Training State
  - Prevents loss of progress from hardware or software failures
  - Enables restart from the last saved state, avoiding costly downtime
- Checkpointing Model States
  - Allows reverting to a previous state if training deviates
  - Facilitates hyper-parameter adjustments for optimal training
- Memory State Preservation
  - Saves GPU memory state, not storage state
  - Distinct from storage snapshots, focuses on active memory dump

Classic LLM Checkpointing – Megatron-LM Deployment Model

GPT-3 175B Parameter Model – Example for 128 DGX Superpod 4 DGX-H100 SUs

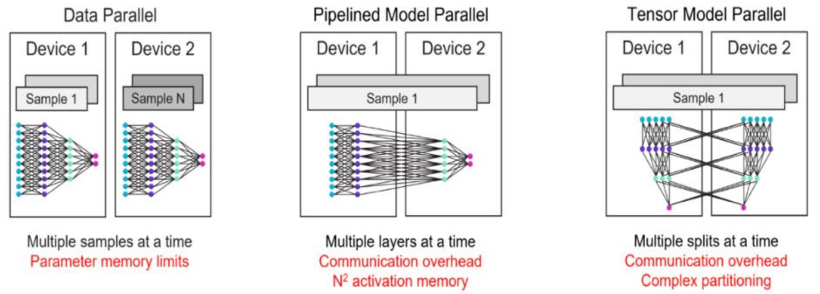


Figure 5 Existing scaling techniques on distributed GPU clusters and their challenges. Scaling on GPU clusters requires a complex combination of all forms of parallelism.

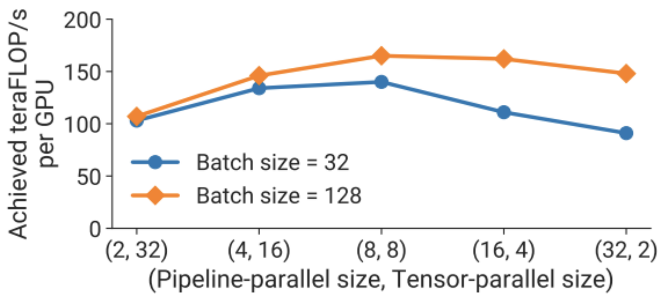


Figure 13: Throughput per GPU of various parallel configurations that combine pipeline and tensor model parallelism using a GPT model with 162.2 billion parameters and 64 A100 GPUs.

- Model Size Exceeds GPU VRAM
  - GPT-3 is ~ 350 GB (2 bytes/parameter)
- Model Is Very Deep (~90 Layers)
- Model Training Is Extremely GPU Intensive

- Tensor Model Parallel
  - Shard Model Across 8 GPUs In A D(H)GX
- Pipeline Parallel set – N DGXs
  - N=16 typically for GPT-3 sized models
- Data Parallel Groups across the pipeline sets

ONLY ONE DATA PARALLEL GROUP of GPUs NEED TO BE CHECKPOINTED
RESTORE NEEDS ALL GPUs TO BE REPOPULATED

https://arxiv.org/pdf/2104.04473.pdf

Input	Value
Checkpoint time (sec)	60
Model Size (B parameters)	7
Checkpoint frequency (sec)	3600
Bytes per parameter	14
Tensor Model Parallelism	4
Pipeline Parallelism	1
Number of GPUs	256
GPU Type	H800
Number of training Tokens (B)	1500
FLOPs per parameter for 1 token	6
FLOP/s/GPU from Megatron paper (petaFLOP/s)*	426
* H100 FLOP/s is roughly 3xA100 FLOP/s from Table	
* Blackwell is estimated to be 3-5x H100 Flops/s	

Parameter	Description	Reference
Checkpoint Time	Completion time in seconds	General guideline
Model Size	Billions of parameters	General guideline
Checkpoint Frequency	Frequency in minutes/hours	General guideline
Bytes per Parameter	14 bytes	Frontier paper (Dash et al. 2023)
Tensor Model Parallelism	Guidelines from Megatron paper	Narayanan et al. 2021
Pipeline Parallelism	From Megatron paper	Narayanan et al. 2021
Number of GPUs	As many as affordable	Table I reference
GPU Type	Determines FLOPs/sec	General guideline
Number of Tokens	Related to Chinchilla Scaling	Discussion below
FLOPs per Parameter	6 FLOPs	Kaplan et al. 2020

MODEL: LLM INPUTS

START WITH A SMALL MODEL (7B)

Number of parameters (billion)	Attention heads	Hidden size	Number of layers	Tensor model-parallel size	Pipeline model-parallel size	Number of GPUs	Batch size	Achieved teraFlOP/s per GPU	Percentage of theoretical peak FLOP/s	Achieved aggregate petaFLOP/s
1.7	24	2304	24	1	1	32	512	137	44%	4.4
3.6	32	3072	30	2	1	64	512	138	44%	8.8
7.5	32	4096	36	4	1	128	512	142	46%	18.2
18.4	48	6144	40	8	1	256	1024	135	43%	34.6
39.1	64	8192	48	8	2	512	1536	138	44%	70.8
76.1	80	10240	60	8	4	1024	1792	140	45%	143.8
145.6	96	12288	80	8	8	1536	2304	148	47%	227.1
310.1	128	16384	96	8	16	1920	2160	155	50%	297.4
529.6	128	20480	105	8	35	2520	2520	163	52%	410.2
1008.0	160	25600	128	8	64	3072	3072	163	52%	502.0

Table 1: Weak-scaling throughput for GPT models ranging from 1 billion to 1 trillion parameters.

Cost ($/GPU-hr)	5
Failure rate (per day/1K GPUs)	0.40
exaFLOP/s available	0.109
yottaFLOP needed	0.063
Token dataset size (TB)	6.00

Parameter	Description	Reference
Cost	Going rate for GPU-hr	Market
Failure Rate	Per day/1000 GPUs	Empirical data
exaFLOP/s available	#GPU x FLOP/s/GPU	From Megatron paper (above)
yottaFLOP needed	6 x model size x #tokens	Kaplan et. Al 2020
Token Dataset Size	4 byte per token	For GPT Style models

COMPUTATIONAL BUDGET CALCULATIONS

FAILURE RATES IN TRAINING
(EMPIRICAL DATA: 0.4-1.2/day/1000 GPUs)

OUTPUT CALCULATIONS:

Output Calculations

Output	Value
Checkpoint size (GB)	4396
Checkpoint impact (% of total time)	3.33%
Checkpoint Write Bandwidth Required (GB/s)	73.3
Checkpoint file size (GB)	68.7
Number of GPUs that checkpoint	64
Write Bandwidth per GPU (GB/s)	1.1
Number of checkpoints per day	48
Total storage required per day (TB)	211.008
Storage for full training (PB)	9.7
Training Time estimate (days)	46.0
Time spent in checkpointing (days)	1.53
GPU Cost for training (Million $)	$22.06
Expected % of runs that will have no failure	0.00%
Expected number of failures during the run	184

GPT-3: A CASE STUDY

- Model Training Parameters
  - 175B model training on 1.7T tokens
  - Impact set at 5%, checkpoint frequency at 5 mins
- Checkpointing Calculations
  - Checkpoint state of 2450 GB for 175B parameters
  - 19.14 GB per GPU for checkpoint files
- Performance Estimation
  - Checkpoint time is 15s (5% of 300s)
  - Required write bandwidth of 163.3 GB/s
  - Checkpoint impact of 5% on performance
  - In practice, 60s checkpoint time is reasonable, with hourly checkpoints => 41.8 GB/s Write Bandwidth
- Checkpoint Interval Considerations
  - Trade-offs between checkpoint frequency and rework costs
- Computational Power and Runtime

Model Training and Checkpointing Parameters

Parameter	Value
Model Size	175B
Tokens	1.7T
Checkpoint State	2450 GB
Checkpoint Frequency	5 mins
Write Bandwidth Needed	163.3 GB/s
Checkpoint Impact	5%
Estimated Runtime	23.14 days

KEY INSIGHTS IN LLM CHECKPOINTING

- Checkpoint Considerations
  - The size of the checkpoint depends ONLY on Model Size
  - NOT on checkpoint time, frequency, number of tokens or number of GPUs used in training
  - Number of tokens and model size drive runtime, with a given FLOP/s budget
- Storage Performance and Costs
  - Required storage capacity and performance metrics
  - Cost implications and optimization strategies
- Deployment Influences
  - Effect of deployment methods on checkpoint strategy
- Comprehensive Analysis
  - This talk provides qualitative and quantitative insights
  - Mathematical model to illustrate tradeoffs and choices

CHECKPOINT INTERVAL CONSIDERATIONS

- Checkpoint Frequency Comparison
  - 15 minutes vs. 5 minutes reduces bandwidth needs by a factor of 3
  - Real-life checkpoints typically range from 1-4 hours
- 5 Minutes vs. 30 Minutes Checkpointing
  - 5% tolerance checkpoint must finish in 90s vs. 15s
  - Total job runtime impact remains the same
- Daily Checkpoint Analysis
  - 288x15s checkpoints or 48x90s checkpoints per day
  - GPUs idle time is consistent regardless of frequency

COST-BENEFIT ANALYSIS OF CHECKPOINTING

- Cost Analysis of GPU Rework
  - 1000 GPUs with 30 mins rework equals 500 GPU-hours
  - Cost estimated at $4/GPU-hour
  - Total rework cost approximates to $2000
- Storage Investment Consideration
  - Potential investment in millions for additional storage
  - Management of increased storage capacity
- Business Decision Tradeoffs
  - Assessing the need for aggressive checkpointing intervals
  - Understanding the tradeoffs in cost and management

STORAGE CAPACITY REQUIREMENTS FOR CHECKPOINTS

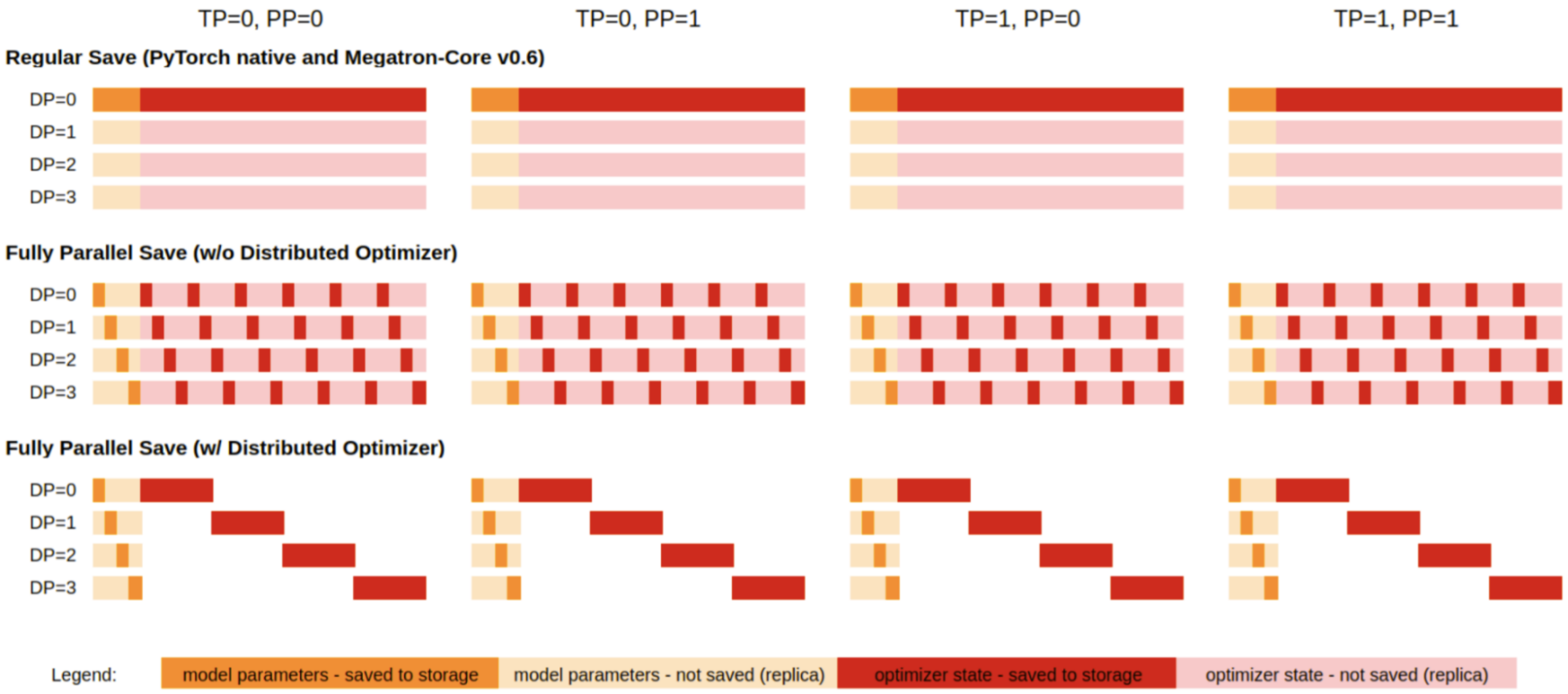
- Checkpointing Frequency and Storage Requirements
  - Checkpointing 2.45 TB every 5 mins requires 163 GB/s Write bandwidth.
  - Job duration: 23 days with 1920 H100 GPUs.
- Daily and Total Storage Calculation
  - 288 checkpoints daily, each 2.45 TB, totaling 0.705 PB/day.
  - Total storage needed for the run: 16.3 PB.
- Operational Challenges and Data Management
  - Checkpoint management is cumbersome during the run.
  - Restoration requires fast storage for all GPUs, not just checkpointed ones.
- Checkpoint Frequency: A Strategic Choice
  - Frequency impacts Write Bandwidth and capacity needs.
  - Balance between cost and value is crucial.

Checkpointing Storage Requirements

Checkpoint Frequency	Write Bandwidth (GB/s)	Daily Storage (PB)	Total Storage (PB)
Every 5 mins	163	0.705	16.3

RECENT DEVELOPMENTS

- Megatron-LM and Megatron-Core 0.7 have introduced async and distributed checkpoints
  - Upstreamed to Pytorch

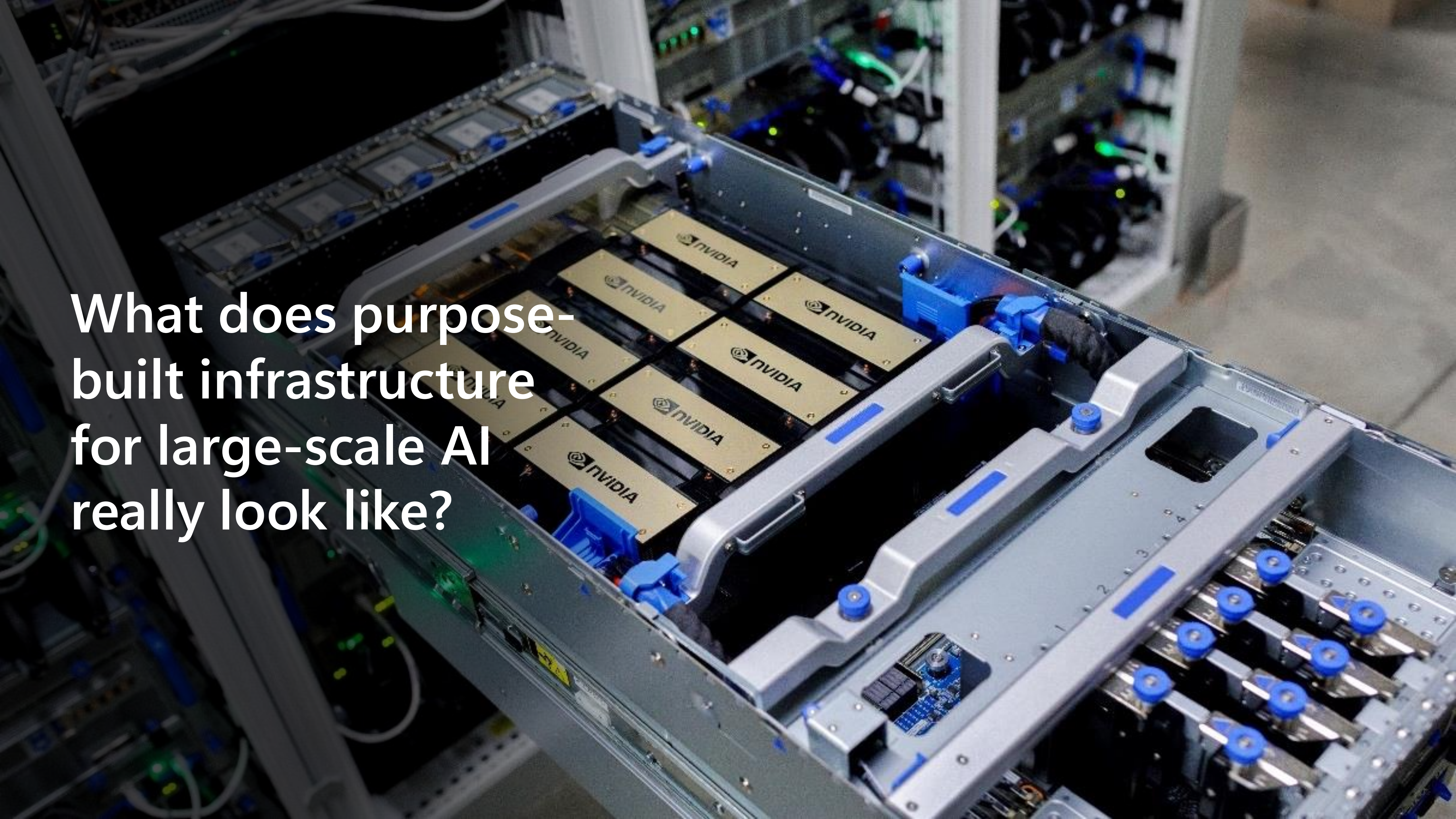


https://developer.nvidia.com/blog/train-generative-ai-models-more-efficiently-with-new-nvidia-megatron-core-functionalities/

REFERENCES

Academic References and Their Online Sources		
Author(s)	Year	Source Link
He et al.	2023	Link
Narayanan et al.	2021	Link
Dash et al.	2023	Link
Kaplan et al.	2020	Link
Hoffmann et al.	2022	Link
Maurya et al.	2023	Link
Wang et al.	2023	Link

Everything Kartik just said is true.


What does purpose-built infrastructure for large-scale AI really look like?
NVIDIA

Inside an Azure NDv5 hardware node

2x 56c Intel Sapphire Rapids	Host CPUs
2.0 TB DDR5-4800	Host DRAM
8x NVIDIA H100 / 80 GB HBM 8x NVIDIA H200 / 141 GB HBM 8x AMD MI300X / 192 GB HBM	GPU options
8x 3.84 TB E1.S NVMe	Local scratch
1x 960 GB M.2 NVMe	Boot disk
2x 1.92 TB M.2 NVMe	Service cache
8x400G NDR InfiniBand	Backend NICs
Microsoft 100G SmartNIC	Frontend NIC

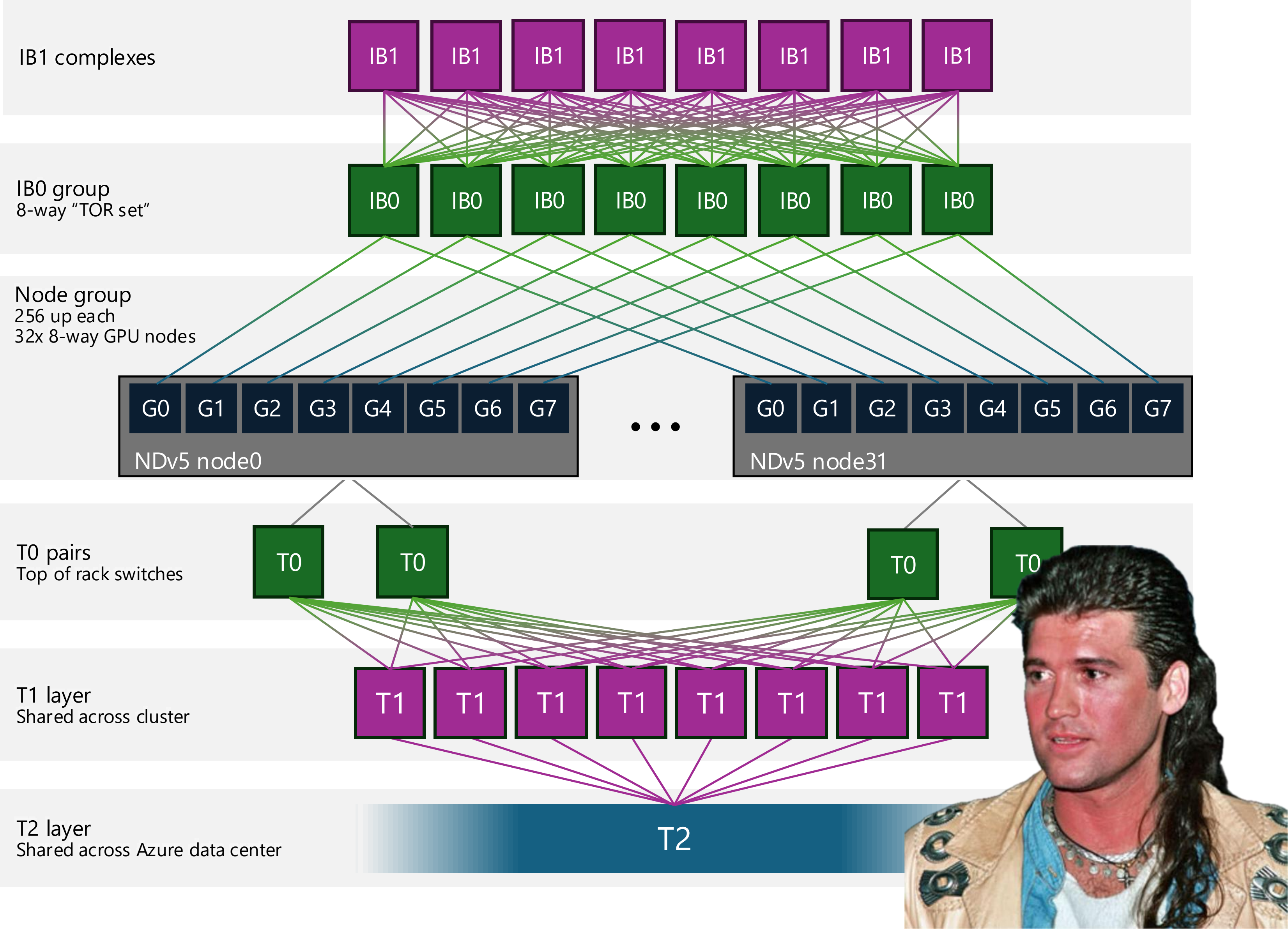
An example NDv5 supercomputer

Backend network

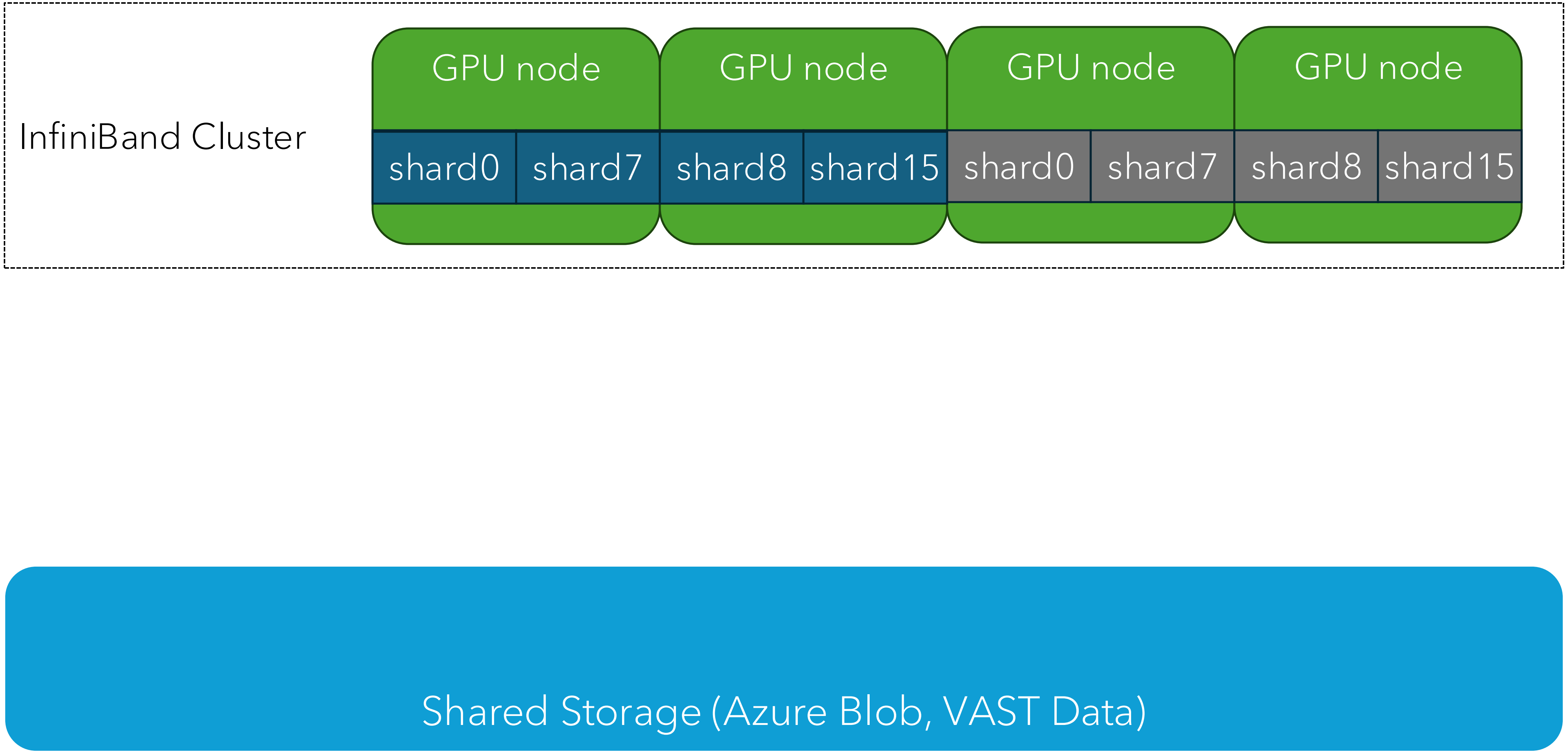
- Non-blocking fat tree
- RDMA (400G NDR)
- NVIDIA ConnectX-7
- No external routes
- Eight planes

Frontend network

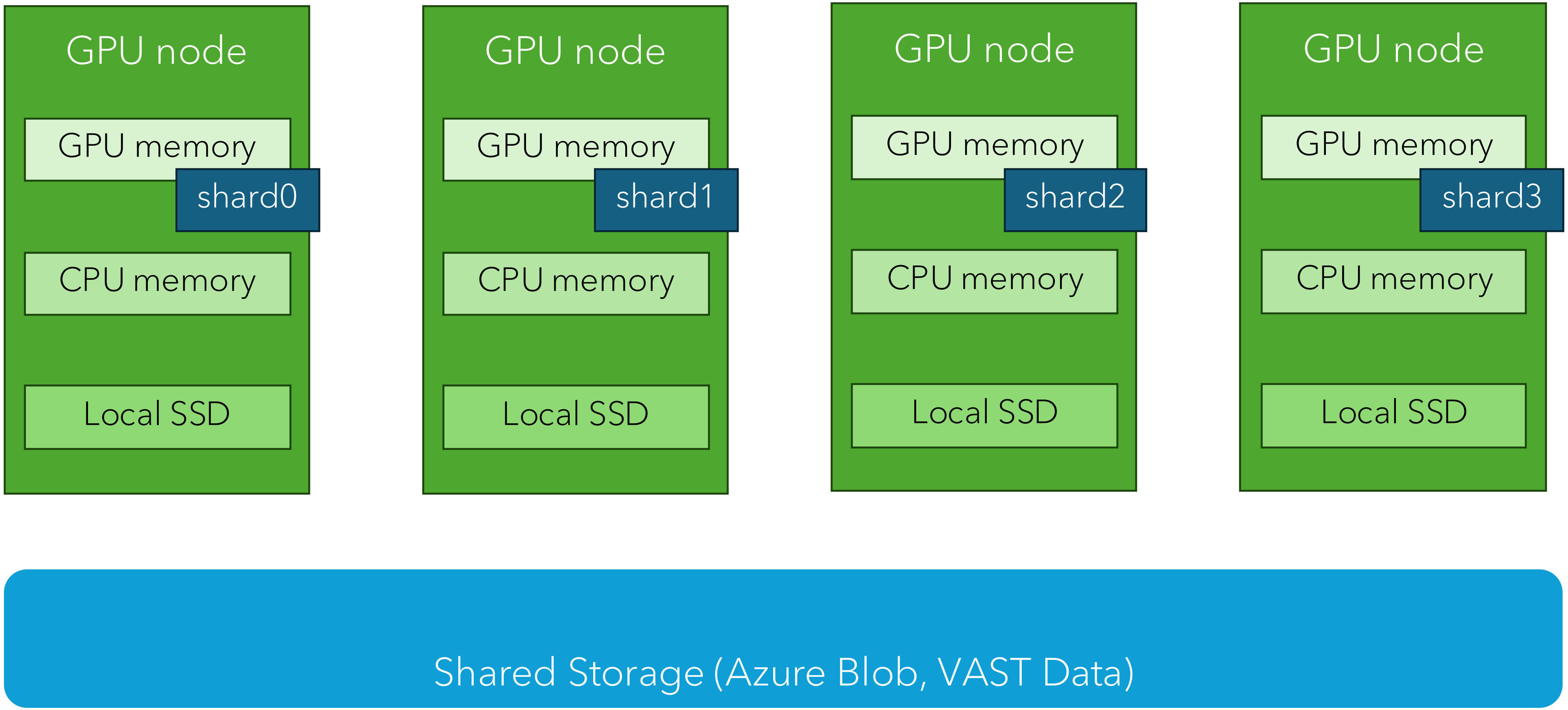
- Tapered
- TCP/UDP (100 GbE)
- Azure SmartNIC
- All N/S traffic (storage, Azure, Internet)
- Fully virtualized



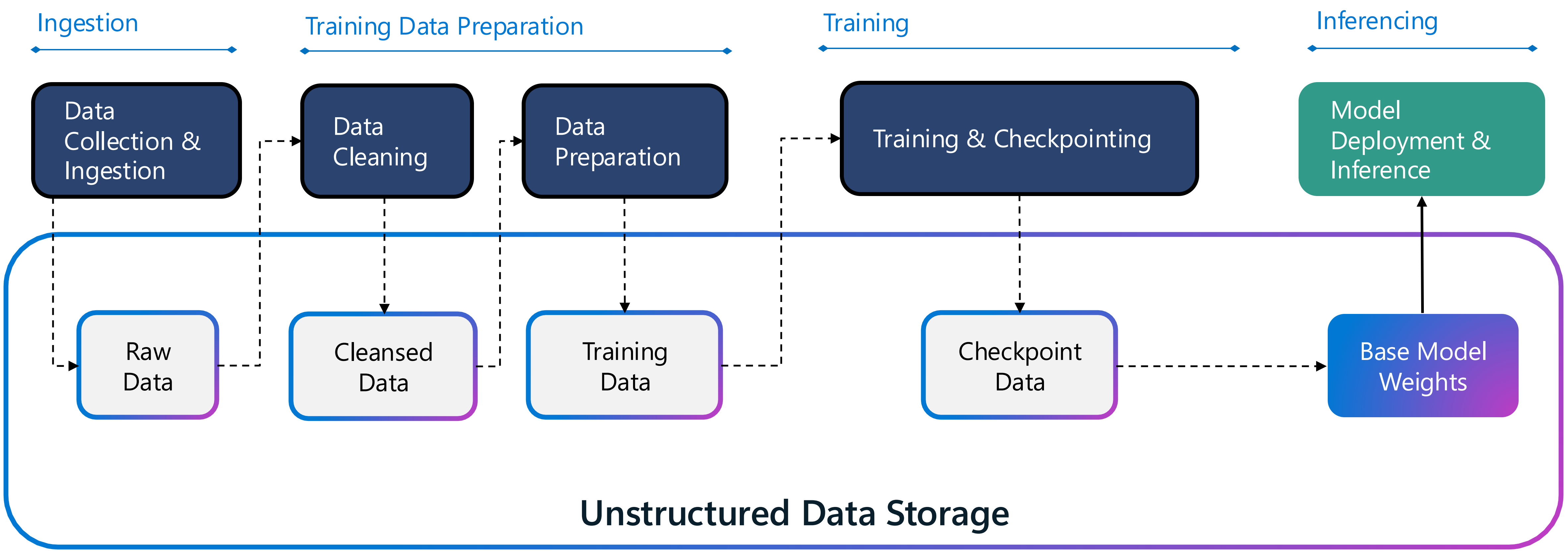
Checkpointing directly to shared storage



Hierarchical checkpointing



AI Pipeline – Storage centric view



CONCLUSION

- We Challenge One-Size-Fits-All Advice
  - Let the data drive what performance and capacity requirements LLMs really need to handle Checkpointing
  - Advocate for decisions based on data, not dogma
- Understanding LLM Behavior
  - Emphasizes calculating LLM behavior from first principles and real data
  - Rejects rationale-less guidance for LLM training requirements