Sizing of AI Clusters: What to Consider

Derrick Monahan
Principal Solutions Architect
AI and High Performance Architecture (HPA)

World Wide Technology

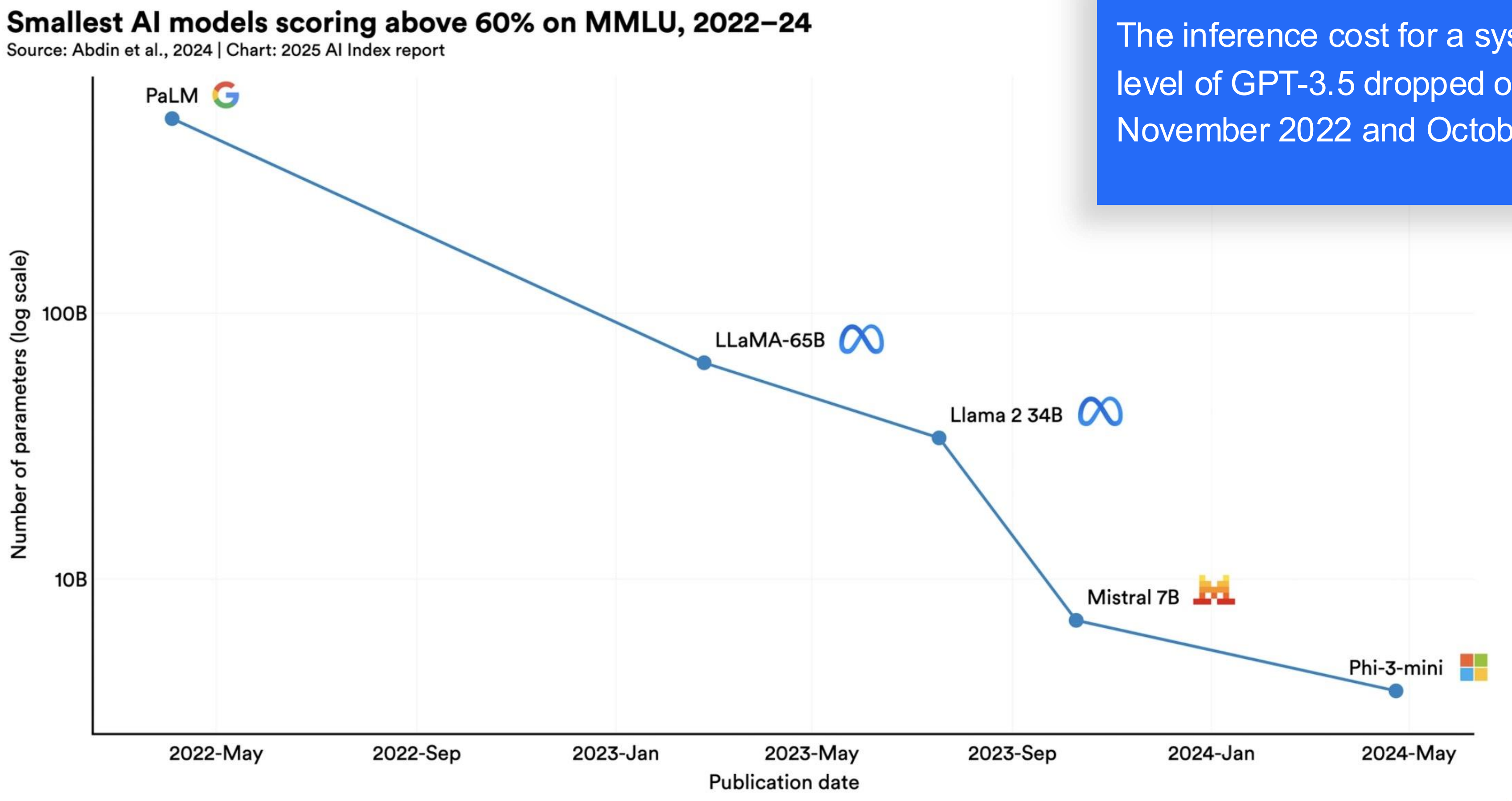
The AI Frontier

Efficiency & Innovation at Scale

The Sizing Challenge

This session addresses the **multifaceted process** and critical **decision points** involved in determining the optimal scale, capacity, and configuration of the infrastructure required to effectively and efficiently train & run artificial intelligence workloads

AI: Driving Greater Efficiency, Affordability, and Accessibility



Smallest AI models scoring above 60% on MMLU, 2022–24

Source: Abdin et al., 2024 | Chart: 2025 AI Index report

The inference cost for a system performing at the level of GPT-3.5 dropped over 280-fold between November 2022 and October 2024

Source: https://hai.stanford.edu/ai-index/2025-ai-index-report

Our Journey Today

- Defining Your AI Use Cases (The Foundation)
- Compute & Memory Considerations
- Storage & Networking Needs (The Backbone)
- Scalability & Future-Proofing
- Cost vs. Performance Trade-Offs
- An AI Sizing Estimation Model (Practical Guidance)

Core Takeaway

AI Cluster Sizing

Key considerations

- **Compute and Memory:**
  - GPUs/CPUs/TPUs/Accelerators – The "brains" for model training speed & inference capacity
  - System RAM & On-Accelerator HBM – Accommodate Large Models and Datasets more efficiently
- **Networking:** Bandwidth & Latency (East-West & North-South)
  - Ultra-low latency, highly scalable, RDMA, Lossless connectivity, dynamic adaptability
- **Storage:** Capacity & Performance (IOPS/Throughput)
  - Fast access to vast datasets & models.
- **Software stack and Orchestration:**
  - AI frameworks, cluster management, compatibility, open-source vs. proprietary

Why Sizing Matters

- Business executive, project manager, etc is building an AI-centric application. They know:
  - What the business goal is
  - What the model(s) are
  - What the application architecture looks like
  - What the user base looks like
- They want to know:
  - What kind of hardware/infrastructure do I need?
  - How much of it do I need?

Four key building blocks for AI Cluster Sizing

Building a High Performance Architecture

Compute

HPC / supercomputing

Accelerated computing

Heterogenous computing

Emergent computing

Quantum computing

Storage

Parallel file system storage

Streaming storage

Data Platforms

Synthetic data

Computational storage

Emergent storage

Network

Connects users and infrastructure

Secure, smart, fast fabrics

SmartNICs and DPUs

Computational networking

Photonics (SOC, switches, backplanes)

Orchestration & AI Workflow

AI and Data Science Tools and Frameworks

Cloud-Native Management and Orchestration

Infrastructure Optimization

Cluster Management

The Criticality of Getting Sizing Right

Performance & Time-to-Value: (Impact on training speed, inference latency, user experience, innovation velocity).

Cost Optimization (CapEx & OpEx): (Avoiding overspending on hardware/power/cooling, maximizing ROI).

Scalability & Future-Proofing: (Adapting to evolving models/datasets, avoiding bottlenecks, strategic growth).

The Perils of Guesswork

A Sizing Story

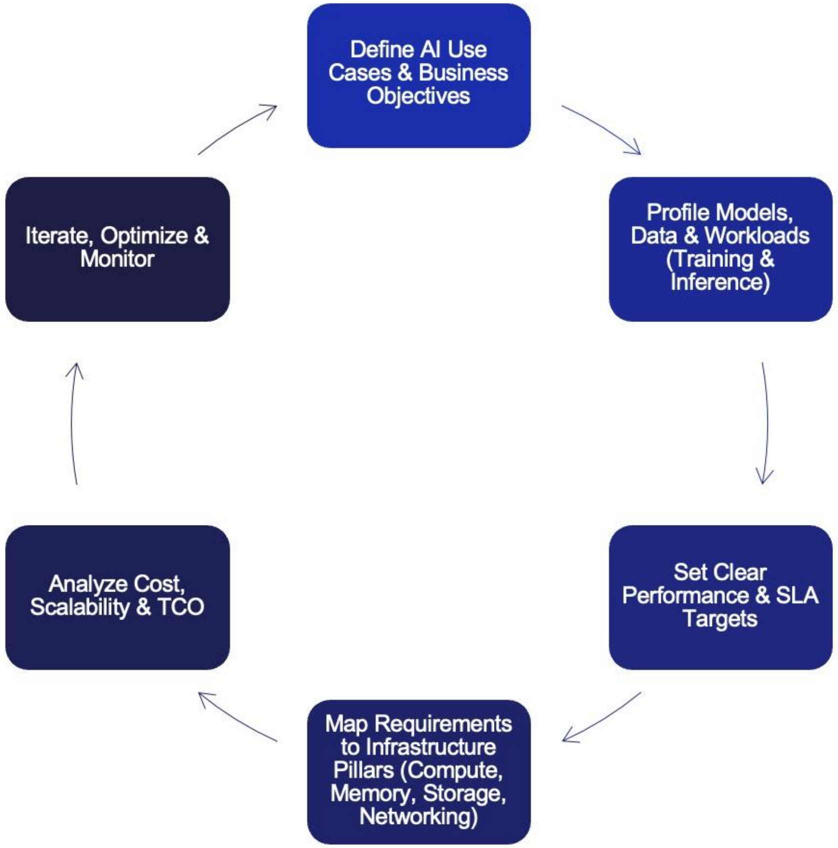
Case Study:

- Large enterprise customer building a GenAI ‘AI Factory’, makes a large GPU investment based on ‘one-size fits all’/generic advice.
- **Training Impact:** Undersized/misconfigured network & storage for specific distributed training needs -> GPU starvation, prolonged training, wasted compute cycles.
- **Inference Impact:** Over-provisioned for actual inference load but geographically centralized -> high OpEx for idle resources + poor latency for global users.

- Key Lesson: Lack of upfront use case definition and holistic sizing

Navigating the Sizing Maze

A Structured Approach



Define AI Use Cases & Business Objectives

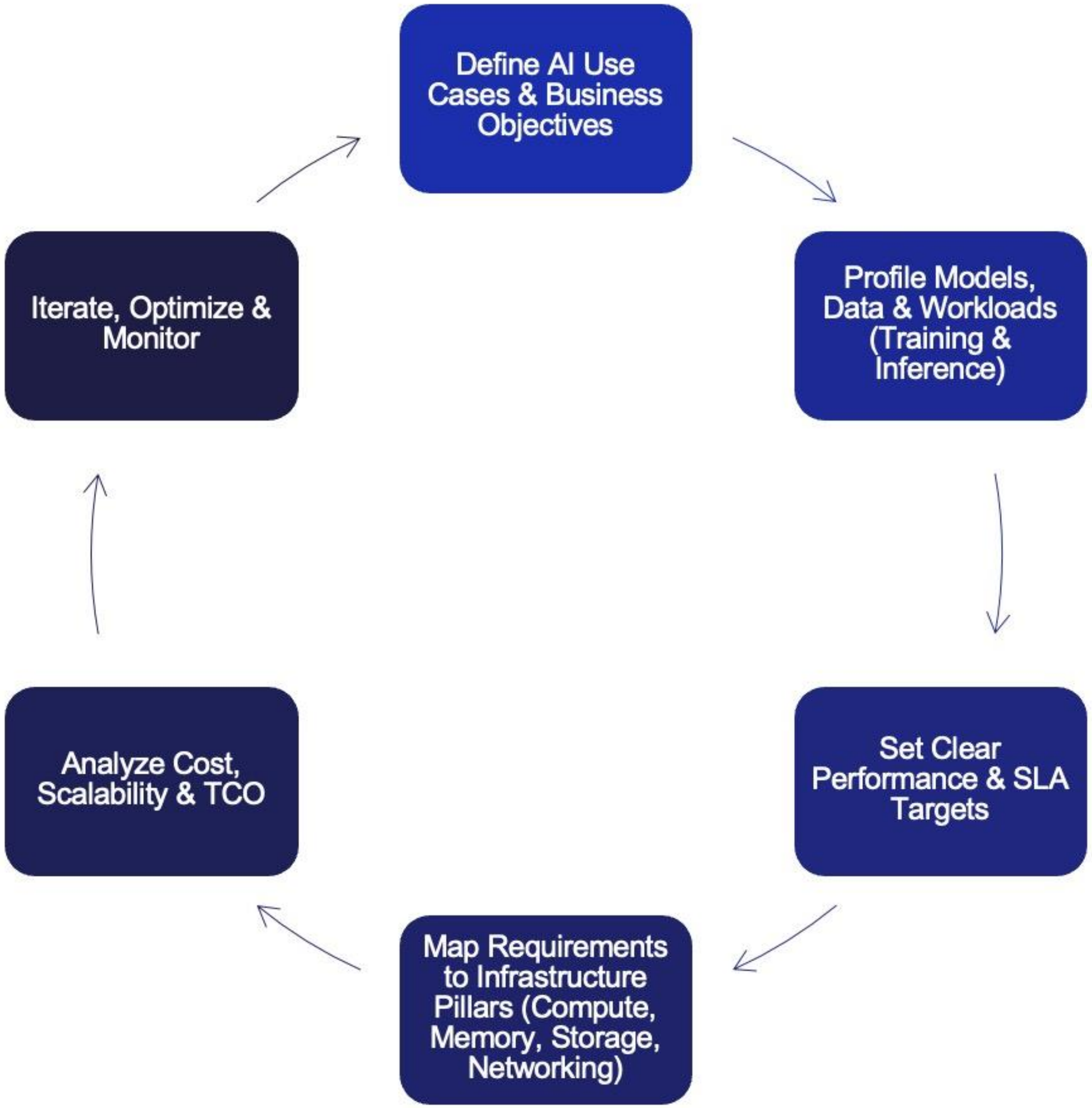
Profile Models, Data & Workloads (Training & Inference)

Set Clear Performance & SLA Targets

Map Requirements to Infrastructure Pillars (Compute, Memory, Storage, Networking)

Analyze Cost, Scalability & TCO

Iterate, Optimize & Monitor



Define AI Use Cases & Business Objectives

Profile Models, Data & Workloads (Training & Inference)

Set Clear Performance & SLA Targets

Map Requirements to Infrastructure Pillars (Compute, Memory, Storage, Networking)

Analyze Cost, Scalability & TCO

Iterate, Optimize & Monitor

Compute & Memory

The Symbiotic Powerhouse

The Evolving Role of Compute & Memory

Workload-Specific Performance: GPU performance can vary significantly depending on the specific AI task (e.g., training a CNN vs. inferencing an LLM).

Sizing must consider the target workloads, not just generic GPU specs.

Specialized Cores: Modern GPUs, (e.g. NVIDIA's H100/H200 successors like Blackwell, and AMD's Instinct MI300 series) feature increasingly sophisticated tensor cores, matrix multiplication units, and even specialized units for things like transformer engines.

CPUs as Orchestrators: Crucial for data pre-processing, managing the OS and AI frameworks, sequential tasks, and, in some cases, offloading parts of the model if GPU memory is exceeded

Integrated CPU+GPU Designs: Emerging integrated solutions (e.g., "NVIDIA's Grace Blackwell Superchip (GB200)," "AMD MI300A APU") that tightly couple CPUs and GPUs with unified memory architectures or very high-bandwidth links (like NVLink-C2C)

System Memory: The ratio of system memory to total GPU HBM is workload-dependent. It's not just about capacity but also bandwidth to avoid CPU-side bottlenecks

Sizing your Compute & Memory: Critical Questions

What are the critical questions you need to answer when sizing compute for your AI cluster?

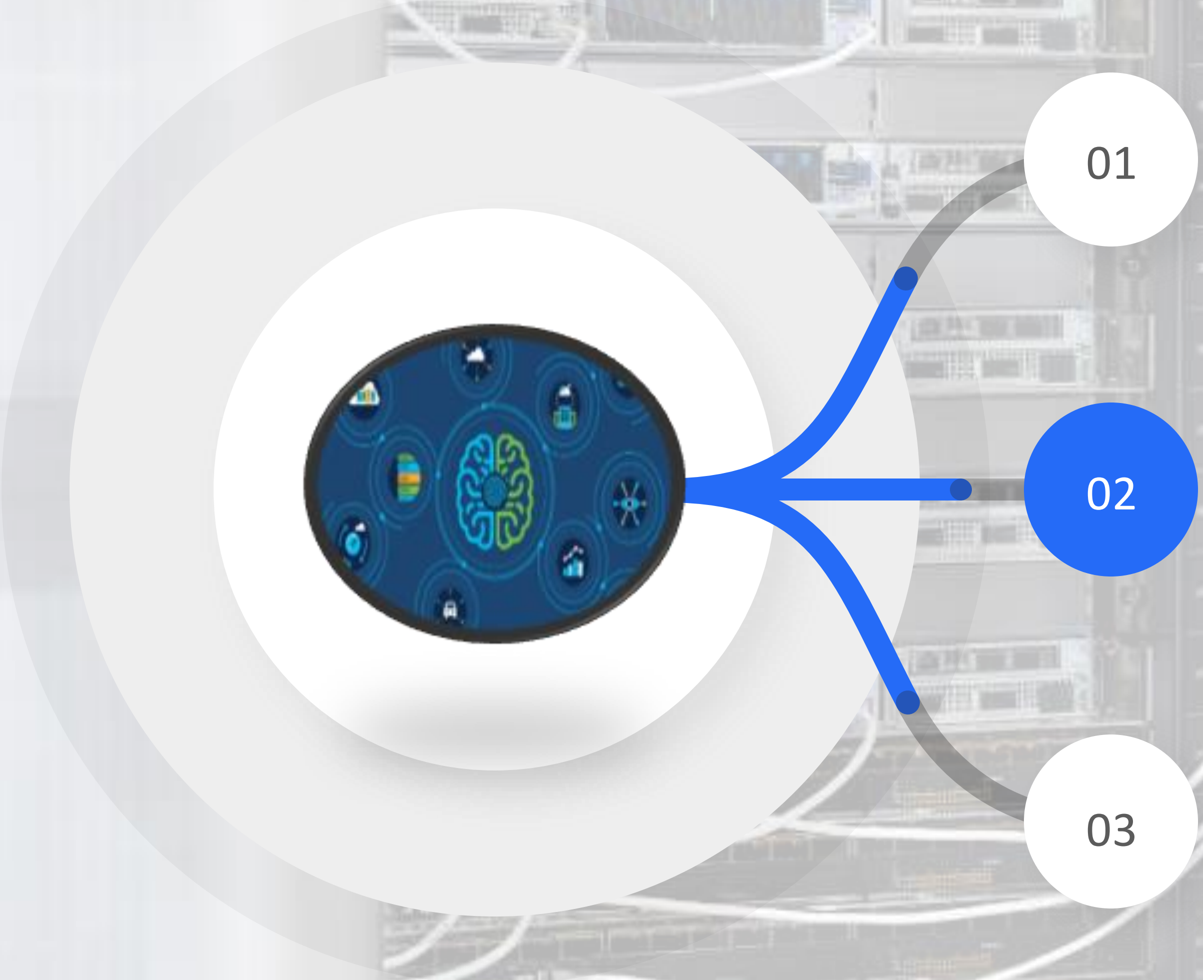
- What are the target AI models (architecture, parameter count)? This directly influences HBM needs.
- What is the batch size for training/inference? Impacts memory per GPU.
- What is the required precision (FP64, FP32, FP16, INT8, FP4)? Affects memory and compute speed.
- Performance Targets: What are the training time targets or inference latency/throughput SLOs?
- How much data needs to be pre-processed by compute, and how fast?
- Will the workload benefit from tightly coupled CPU-GPU memory (e.g., Grace Blackwell)?

Igniting AI Clusters

Unleashing the Power of High-Performance Storage and Intelligent Networking

Beyond the GPUs:

Storage & High-Performance Networking Are the Unsung Heroes (and Often Overlooked) in AI Infrastructure



01

AI Clusters Require Lossless Networks

Parallel Processing Sensitivity
Data Integrity for Model Accuracy
Predictable Low Latency and Jitter

02

Network automation and AI/ML

Lower the cost of network operations
Provide users with an optimal connected experience

03

AI/ML and telemetry

Visibility into Network Behavior
Improves Transport

High-Performance Networking

Considerations for AI

- RDMA is King (InfiniBand and RoCEv2):
  - **RoCEv2** (RDMA over Converged Ethernet), InfiniBand (mature RDMA implementation)

- Key Network Metrics & Design
  - **Tail Latency:** Beyond average latency, minimizing *tail latency* (the latency experienced by the slowest packets) is crucial because distributed training jobs often wait for the slowest worker (straggler).
  - **Effective Bandwidth at Scale**
  - **Congestion Control**

- In-Network Computing/Acceleration:
  - **SmartNICs/DPUs:** important to offload collective communication operations (like reductions) or parts of the data pipeline directly into the network, reducing load on CPUs/GPUs.

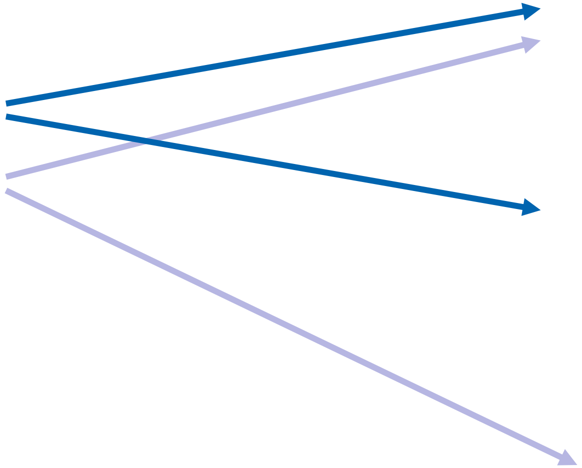
High-Performance Networking:

AI Communication Patterns

AI communication patterns are fundamental to sizing AI clusters, particularly the network infrastructure.

Common AI Collective Communication Patterns:

- All-Reduce
- All-to-All (or A2A)
- Scatter
- Reduce
- Gather



Why These Patterns Are CRITICAL for AI Cluster Sizing

Bandwidth Demand:

- All-Reduce and All-to-All are bandwidth-hungry.
- Sizing Implication: Your network links

Latency Sensitivity:

- Distributed Training involves many iterations. An All-Reduce for gradient synchronization might occur multiple times
- Sizing Implication: Low-latency interconnects

Scalability Challenges:

- As nodes increase, communication overhead becomes the bottleneck
- Sizing Implication: Scaling dynamics: Sizing must account for the target scale and how these patterns behave at that scale

Sizing your Network: Critical Questions

- What are your dominant AI communication patterns (e.g., All-Reduce heavy for data parallelism, All-to-All for model parallelism)?
- What are your latency and bandwidth targets per GPU and per job?
- What is your desired level of network oversubscription, especially in the face of bursty AI traffic?
- Which RDMA technology (InfiniBand or meticulously planned RoCEv2) best fits your scale, budget, and expertise?
- How will your network topology (e.g., Rail-optimized full fat-tree) efficiently support these patterns at your target scale?

High-Performance Storage Considerations for AI

- Throughput for Training:
- IOPS & Latency for Metadata & Small Files:
- Checkpointing Performance:
- Data Loading & Caching:
- Data Formats & Access Patterns:
  - Understanding the read/write ratio, sequential vs. random access patterns, and average file sizes of your specific AI workloads is crucial for selecting and sizing the right storage solution

Planning for Tomorrow's AI

Scalability & Future-Proofing

Scalability & Future-Proofing

AI doesn't stand still. How do we size and design clusters not just for today's needs, but for the rapidly evolving demands of tomorrow?

Scalability Imperative in AI

Meeting Exponential Growth

Larger models (trillions of parameters are here!), bigger datasets, more users, new AI applications, expanded use cases

- **Scale-Up vs. Scale-Out:** Understanding the approaches and their implications for AI.
- **Modular Design:** Building with standardized "building blocks" or "pods" for predictable expansion.
- **The Unseen Hurdles:** Power, cooling, and physical space – critical, often underestimated, scaling limiters.

Future-Proofing

Designing for an Evolving AI Landscape

Embrace Architectural Flexibility:

- Design for a mix of AI workloads (training, inference, diverse model types like MoE, multimodal)

Anticipate Hardware Evolution:

- Plan for GPU/accelerator refresh cycles.
- Consider disaggregated infrastructure concepts

Network Agility:

- Ensure your network fabric can support next-gen speeds & adapt to new communication patterns

Orchestration and MLOps:

- Leverage software for infrastructure management, orchestration (Kubernetes is key here), and resource abstraction

The Hybrid Advantage:

- Strategically using public cloud for burst capacity or access to specialized hardware not yet on-prem

Scale/Future-Proof AI: Critical Questions

- What is your **projected growth** in model size, dataset volume, and user load over the next 1-3 years?
- How will your chosen network and storage **architectures scale** to meet this growth without performance degradation or excessive cost?
- What is your strategy for incorporating new AI accelerator technologies as they become available?
- How much headroom are you building into your power, cooling, and physical space planning?
- Does your software and orchestration layer support **heterogeneous hardware** and **dynamic resource allocation** to adapt to future needs?

The Balancing Act

Tackle the crucial trade-offs between cost and performance.

Where Performance Meets Budget: Key Trade-Off Arenas

- **Compute:**
  - Latest-gen GPUs (premium cost) vs. prior-gen (better price/perf for some workloads)? Density (more power/cooling per rack) vs. spreading across more nodes?
- **Networking:**
  - InfiniBand (highest performance) vs. meticulously engineered RoCEv2 (potential cost savings, higher complexity)? Level of fabric oversubscription? 400G now vs. readiness for 800G/1.6T?
- **Storage:**
  - All-flash parallel systems (peak performance) vs. tiered storage with intelligent caching (balanced cost/performance)? Capacity of local NVMe on compute nodes?
- **Memory:**
  - Maxing out HBM on GPUs and system RAM (future-proofing for larger models) vs. "just enough" for current targeted workloads (potential near-term savings, future risk).

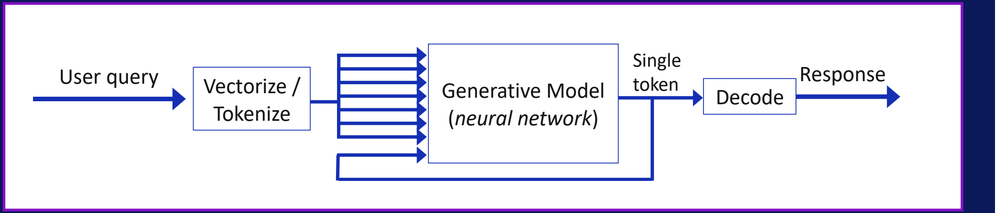
An AI Sizing Estimation Model

Practical Guidance

Why Sizing Matters

- Business executive, project manager, etc is building an AI-centric application. They know:
  - What the business goal is
  - What the model(s) are
  - What the application architecture looks like
  - What the user base looks like
- They want to know:
  - What kind of hardware/infrastructure do I need?
  - How much of it do I need?

AI Performance & Sizing Principals for Inferencing



Is the workload *Real-Time* or *Batch* ?

Batch is cheaper: uses less compute overall

The lower the latency the more GPUs required – Tensor Parallelism

Volume of Requests (per second or per minute)?

More requests per second the more GPUs needed (*10k per minute vs 10k per hour*)

Size of Model?

Number of parameters x2 = **size on GPU in GB** (FP16)

Type of Model?

Each model has a diff way it is implemented on the GPU leading to diff performance. (*logistic regression vs RNN vs transformer, etc.*)

Type of Hardware?

Hopper GPUs have FP8 (reduces mem by half)

Blackwell GPUs have FP4

Llama2 (7B) – 14GB (FP16) **but in FP8 = 7GB**

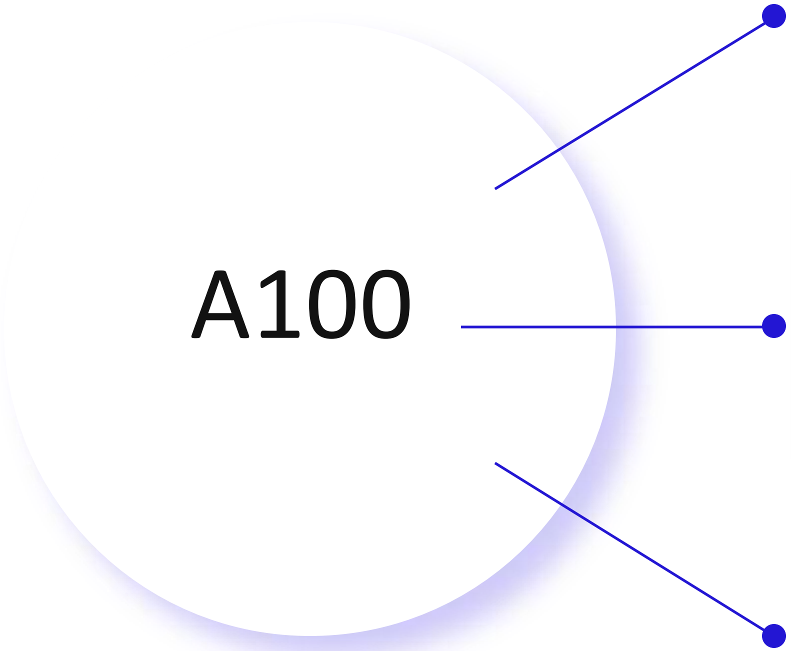
Avg size of input and output prompts for the use case (tokens)

Understanding the **cost dynamics** of LLM Inference – think about the use cases (4 common ones discussed)

Prompt size affects memory usage

Basic Concurrency Example:

A100 FP16 Sizing for **7B** Model Inference



Key Parameters

- **Model**: 7B parameters, FP16 (~14 GB memory)
- **Workload**: 100 tokens/query, 30 queries/second, 15 tokens/second/user
- **Total Throughput**: 3000 tokens/second

GPU Estimate

- **1 A100 GPUs needed**
- Memory: ~14 GB (weights) + overhead
- Tensor Parallelism: 2 GPUs/instance (~40 tokens/second)
- Concurrency: Non-linear scaling for 30 queries
- Infrastructure: 1DGX servers, 6.5kW power

Takeaways

- Use TensorRT-LLM and batch size 32 to optimize performance.
- Validate with WWT’s AI Proving Grounds for high concurrency.

Large Concurrency Example: 5000 queries/sec

A100 FP16 Sizing for **13B** Model Inference

A100

Key Parameters

- **Model**: 13B parameters, FP16 (~26 GB memory)
- **Workload**: 500 tokens/query, 5,000 queries/second, 25 tokens/second/user
- **Total Throughput**: 2.5M tokens/second
- **Performance**: ~25 tokens/second/GPU (batch size 64, TensorRT-LLM)

GPU Estimate

- **8,000 A100 GPUs needed**
- Memory: ~26 GB (weights) + ~0.08 GB/query (KV cache)
- Tensor Parallelism: 2 GPUs/instance (~25 tokens/second)
- Concurrency: Non-linear scaling for 5,000 queries
- Infrastructure: ~1,000 DGX servers, ~6.5 MW power

Takeaways

- Use TensorRT-LLM and batch size 64 to optimize performance.
- Validate with WWT’s AI Proving Grounds for high concurrency.
- Consider H100 for ~4x fewer GPUs if budget allows.

Summary: Practical Implications for Sizing

Start with the Model(s) and Data:

How large are the models? What precision will be used? How much memory per parameter is needed for weights, activations, optimizer states (for training)? How large is the KV cache (for inference)?

Determine Per-GPU Capacity:

Based on the above, can the model (or a shard of it, plus associated data) fit into the memory of your target GPU (e.g., 192GB on a Blackwell B200 or MI300X)?

CPU to Support GPUs:

Ensure CPUs have enough cores and system RAM to efficiently preprocess data and manage the GPUs without becoming a bottleneck.

Consider the Entire Workflow:

Sizing isn't just for the core training loop or inference execution but the entire end-to-end pipeline.

Thank You!

Questions

World Wide Technology