FRONT MATTER · CHAPTER 00

Preface, Table of Contents & How to Use This Book

Who This Book Is For · Reader Paths · 12-Month Curriculum · Confidence Labels · Full Table of Contents

Book: AI/ML Infrastructure from Silicon to Scale — V2 Edition | **Author:** Venkat Vinjam | **GitHub:** vvinjamu.github.io/silicon-to-scale

> *"The engineer who can walk into an architecture review and say: 'this design is memory-bandwidth-bound at these shapes — here is the roofline, here is the gap, here are two approaches to close it with their TCO implications' — that is the principal. This book builds that engineer."*
>
> — Book Mission Statement

Preface — Who This Book Is For, and How to Use It

This book began as a living preparation system for one of the most technically demanding roles in modern software: **Principal Performance Architect for AI/ML infrastructure**. It grew through real interview cycles, production architecture reviews, and hands-on performance engineering at GPU cluster scale into a comprehensive reference that covers the full technical stack — from the physics of DRAM cell refresh to the economics of a 10,000-GPU fleet.

Every other resource covers one layer. Deep learning textbooks cover model theory without hardware. CUDA programming guides cover kernel optimization without system context. MLOps books cover pipelines without silicon understanding. Interview prep books give patterns without first-principles math. This book is the only single reference that unifies all of these into a coherent analytical system.

Target Reader

Who will get the most from this book

A strong senior performance engineer — CPU, HPC, cloud, embedded, or networking background — who understands bottleneck analysis, roofline thinking, and measurement discipline, and needs the domain knowledge to apply all of that at principal level in AI/ML infrastructure.

The book serves three reader levels simultaneously:

Level	Background	What You Get
Foundational	Strong engineer new to AI infra	First-principles mental models, "why does this matter" framing, intuitive analogies, no assumed GPU/ML background
Intermediate	Senior engineer, 3–5 years in cloud/HPC/ML	Quantitative depth, cross-system connections, profiling workflows, production numbers you can verify on your own cluster
Principal	Staff/principal engineer, architecture decisions	Cost-performance tradeoff math, system design patterns at scale, failure mode reasoning, ROI-driven decision frameworks, interview-ready depth

Reader Paths — Four Ways Through This Book

Path	Goal	Chapter Sequence
Hardware Track	Deep GPU/accelerator mastery	Ch 1 → 3A → 3B → 4 → 5 → 7 → App A
Inference Track	Production LLM serving expertise	Ch 1 → 2 → 6 → 11 → 8 → 13 → 17
Training Track	Large-scale training systems	Ch 1 → 2 → 10 → 14 → 15 → 16 → 12
Career Track	Principal-level interview readiness	Ch 1 → 18 → 12 → then any deep-dive chapter

The 12-Month Learning Curriculum

```
MONTHS 1-3 (FOUNDATIONS):
  Week 1-4:  Chapter 1 — Performance mindset, roofline, three regimes, MFU
  Week 5-8:  Chapter 2 — Transformer architecture as compute/memory problem
  Week 9-12: Chapters 3/3A/3B — GPU silicon, accelerator landscape, roadmap

MONTHS 4-6 (TECHNICAL DEPTH):
  Month 4:   Chapter 4 (Memory + FlashAttention) + Chapter 7 (Kernels)
  Month 5:   Chapter 8 (Quantization) + Chapter 9 (Fusion/Compilers)
  Month 6:   Chapter 5 (Thermal/TCO) + Chapter 17 (Telemetry/Profiling)

MONTHS 7-9 (SYSTEMS):
  Month 7:   Chapter 6 (Inference) + Chapter 11 (KV Cache) + Chapter 13 (Spec Dec)
  Month 8:   Chapter 10 (Training) + Chapter 14 (Networking) + Chapter 16 (Cluster)
  Month 9:   Chapter 15 (Storage/IO) + Chapter 12 (Benchmarking)

MONTHS 10-12 (INTEGRATION):
  Month 10:  Chapter 18 (Career + System Design) + Chapter 19 (Resilience)
  Month 11:  Chapter 20 (Future) + Chapter 21 (Case Studies)
  Month 12:  All appendices + mock interviews + blog post or GitHub publication
```

Confidence Label System

This book uses confidence labels on all hardware specifications. Technology moves faster than publication cycles — a label tells you whether to trust a number at face value or verify it against current documentation before using it in a design decision.

Label	Meaning	Trust Level
[SHIPPED]	Measured on production hardware; verified against manufacturer datasheets	High — use in production calculations
[ANNOUNCED]	Vendor-disclosed, not yet generally available or shipping	Medium — verify before committing to design
`[ESTIMATED]`	Calculated from first principles or extrapolated from known data	Medium — check the derivation, not just the value
`[REPRESENTATIVE]`	Illustrative example; not a specific product claim	Conceptual — don't use in procurement decisions
`[ENV-SPECIFIC]`	Highly cluster- and framework-dependent; will vary in your environment	Directional — always measure in your system

Table of Contents — All 21 Chapters

Part	Chapter	Title	Key Topics
I · Foundations	Ch 1	Performance Mindset	Seven-layer stack, roofline, three regimes, MFU/HFU
	Ch 2	Transformer Architecture	Attention FLOPs, GQA/MQA/MLA, MoE, full forward pass budget
	Ch 3	GPU Architecture	SMs, Tensor Cores, HBM, NVLink — the hardware model
	Ch 3A	Accelerator Landscape	TPU, Trainium, Gaudi 3, Groq, MTIA — when each wins
	Ch 3B	GPU Generation Roadmap	H100→GB300, MI300X→MI455X, TCO comparisons
II · Physics	Ch 4	Memory Hierarchy + FlashAttention	Memory wall, HBM architecture, FA v1/v2/v3 derivation
	Ch 5	Thermal, Power & TCO	Cooling, TDP, $/token/day, liquid cooling transition
III · Inference	Ch 6	Inference System Architecture	Prefill/decode, PagedAttention, disaggregation, SLA
	Ch 7	GPU Kernels & GEMM	Tiling, Tensor Core alignment, Triton, CUTLASS
	Ch 8	Quantization	FP8 recipes, INT4/AWQ, KV quant, MXFP
	Ch 9	Fusion & Compilers	torch.compile, CUDA Graphs, TensorRT, XLA
IV · Training	Ch 10	Distributed Training	4D parallelism, ZeRO, RLHF/PPO, MoE training
	Ch 11	KV Cache	PagedAttention, radix attention, offloading, eviction
	Ch 12	Benchmarking	MFU/HFU, MLPerf, TTFT/TPOT curves, regression CI
V · Scale	Ch 13	Speculative Decoding + MoE	Draft-verify loop, acceptance rate, expert routing
	Ch 14	Networking	NVLink, InfiniBand NDR, NCCL, AllReduce algorithms
	Ch 15	Storage & I/O	NVMe, Lustre, DALI, checkpoint cadence
	Ch 16	Cluster Management	SLURM, Kubernetes, gang scheduling, SDC, MIG
	Ch 17	Telemetry & Tools	Nsight, DCGM, Prometheus/Grafana dashboards
VI · Principal	Ch 18	Career & System Design	STAR+Impact, ChatGPT system design walkthrough
	Ch 19	Production Resilience	Fault tolerance, SDC, recovery, fleet hardening
	Ch 20	Future of AI Compute	Chiplets, optical interconnects, 2030+ roadmap
	Ch 21	Real-World Case Studies	70B, 405B, multimodal, MoE fleet deployments

Code Block Conventions

```
FORMULA BLOCKS (dark background) — used for:
  • Mathematical formulas with derivation steps
  • Hardware specifications with derived calculations
  • Pseudo-code that illustrates an algorithm at architecture level
  • Quick-reference tables of key production numbers

  Example:
    MFU = (FLOPs_per_step × steps_per_second) / Peak_TFLOPS
        = (2 × N_params × seq_len × batch_size × steps_s) / 989e12

    H100 ridge point = Peak_TFLOPS / HBM_BW = 989 / 3.35 = 295 FLOP/byte
```

Cross-reference tags

— Throughout the book you will see tags like Ch 4: Memory wall and Ch 11: KV paging. These indicate that the concept is covered in depth in the referenced chapter. Cross-references are bidirectional — each chapter links forward and backward to build a connected knowledge graph.