O'REILLY®

Hands-On LLM Serving and Optimization

Hosting LLMs at Scale

Chi Wang & Peiheng Hu

"As LLMs become core to modern AI platforms, this book offers a practical roadmap for serving and optimizing LLMs at scale—handling large workloads with reliability and cost efficiency."

Bhavesh Doshi
Vice president of AI Cloud, Salesforce

Hands-On LLM Serving and Optimization

Large language models (LLMs) are the reasoning engines of modern AI. Today, a major inflection point has arrived: as the world races to deploy AI at scale, model inference has moved to the center of the stack. Welcome to the inference era.

Without proper optimization, however, LLMs can be expensive and slow to serve. *Hands-On LLM Serving and Optimization* is a comprehensive guide to the complexities of deploying and optimizing LLMs at scale.

In this hands-on, engineering-focused book, authors Chi Wang and Peiheng Hu combine practical examples, code, and strategies for building robust, performant, and cost-efficient AI token factories. Whether you're building the LLM inference infrastructure or the applications that consume it, a deep understanding of LLM serving will make you a more effective, future-ready engineer as AI transforms how we work and build.

- **Learn the foundations of model serving with core concepts, design paradigms, and industry best practices**
- **Understand the common challenges of hosting LLMs at scale**
- **Balance latency and throughput to meet the demands of AI applications and business requirements**
- **Host LLMs cost-effectively with practical, code-backed techniques**

Chi Wang is a director of engineering at Salesforce's AI group, where he leads model inference and data science platforms at scale. He has over 18 years of experience in AI and distributed systems, holds a dozen patents, and writes about building practical AI systems.

Peiheng Hu is an LLM inference engineer at NVIDIA, pushing the boundaries of LLM inference performance on the latest GPU architectures. He holds degrees from Harvard and Georgia Tech and has spent over a decade building large-scale distributed AI systems across NVIDIA, Salesforce, and Microsoft.

DATA

US $79.99 CAN $99.99
ISBN: 979-8-341-62149-7

9 798341 621497 57999

O'REILLY®

Hands-On LLM Serving and Optimization

Hosting LLMs at Scale

Chi Wang and Peiheng Hu

O'REILLY®

Hands-On LLM Serving and Optimization
by Chi Wang and Peiheng Hu

Published by O'Reilly Media, Inc., 141 Stony Circle, Suite 195, Santa Rosa, CA 95401.

O'Reilly books may be purchased for educational, business, or sales promotional use. Online editions are also available for most titles (*http://oreilly.com*). For more information, contact our corporate/institutional sales department: 800-998-9938 or *corporate@oreilly.com*.

Acquisitions Editor: Nicole Butterfield
Development Editor: Sarah Grey
Production Editor: Beth Kelly
Copyeditor: Charles Roumeliotis
Proofreader: Laura K. Miller
Indexer: Sue Klefstad
Cover Designer: Susan Brown
Cover Illustrator: José Marzan Jr.
Interior Designer: David Futato
Interior Illustrator: Kate Dullea

May 2026: First Edition

Revision History for the First Edition
2026-04-27: First Release

See *http://oreilly.com/catalog/errata.csp?isbn=9798341621497* for release details.

979-8-341-62149-7

[LSI]

Table of Contents

Preface

Large language models (LLMs) have gone from research curiosities to production-critical infrastructure in a shockingly short time—much like the internet revolution. An agentic world is coming, and in many ways it's already here: a new wave of "tokenization" where more and more applications are built on top of LLM infrastructure rather than traditional APIs and services.

In just a few years, "just call the API" from public LLM providers like OpenAI has evolved into "we need our own models," and then into "we need to run these models efficiently, safely, and at scale." Businesses now need far more control over their LLMs—for data governance, troubleshooting, evaluation, compliance, and cost management. Many teams have discovered that the hardest part of GenAI isn't training a model or wiring up a chat UI—it's everything in between: setting up model serving and optimization that can meet business goals at an acceptable cost.

We've watched that gap up close. We've seen brilliant prototypes crumble under real traffic or blow through a GPU budget in a week. We've seen organizations that are eager to rebuild key use cases for LLMs held back by concerns about public API costs and data safety. We've seen teams that want to embed LLMs deeply into core products but feel intimidated by the complexity: how to reason about latency, throughput, and cost or how to choose between public vendors, model serving libraries, cloud endpoints, or another self-managed service.

At the same time, knowledge about LLM serving and optimization is scattered across blog posts, research papers, framework documentation, and informal production war stories. The domain evolves weekly or monthly; it's hard to keep up, and even harder to know where to start. What's missing is a systematic foundation: a practical, end-to-end resource that helps you understand the core ideas so you can keep exploring as the ecosystem changes.

That's the book we set out to write.

Why LLM Serving and Optimization?

At a distance, LLM serving can look like the next step after classic machine-learning deployment. But in practice, LLMs are unusual beasts. They present a fundamentally different problem with new physics, new economics, and new stakes—and that's why they deserve their own discipline.

Traditional machine learning (ML) models are typically stateless, bounded, and predictable. You send an input, run a fixed computation graph, and get a result. Latency is stable, memory needs are known, and scaling usually just means adding more replicas.

LLMs are different in every meaningful way. They are autoregressive and stateful, generating tokens step by step while maintaining a growing memory of the conversation. They operate in distinct prefill and decode phases that stress hardware differently, and they demand enormous GPU memory and bandwidth. Performance is no longer "how fast a model runs once," but how well you schedule thousands of variable-length conversations in parallel without breaking latency expectations.

Usage is different, too. Classic ML-powered ranking, classification, or risk scoring often supports background decision making. But LLMs sit directly inside interactive user experiences: conversational assistants, reasoning systems, retrieval-augmented generation (RAG) pipelines, and autonomous agents. Latency is visible to users. Streaming isn't optional. Reliability defines trust. Serving is no longer the infrastructure behind a product; it *is* the product experience.

The business impact is correspondingly higher. When an LLM system slows, fails, or behaves unpredictably, entire workflows stall. Agents stop acting, employees lose confidence, and customers churn. Accuracy, guardrails, and observability are not academic—they are operational, financial, and sometimes legal concerns.

And then there is the cost. In classic ML, inference is usually cheap, and in many cases GPU isn't necessary at all. With LLMs, inference *is* the dominant cost. GPU memory becomes strategic. Inefficient scheduling translates directly into wasted dollars. API-only approaches become expensive at scale, yet many teams are intimidated by self-hosting because they don't know how to balance throughput, latency, and cost.

Finally, the serving patterns themselves are new. Continuous batching, token schedulers, key-value (KV) cache management, quantization strategies, model routing, and hybrid pipelines of retrieval, reasoning, and tool execution simply did not exist in prior generations of ML systems. Teams often know *what they want to build*, but not *how to build it well.*

That is why LLM serving and optimization deserve focused treatment. If you are building real systems, you need more than API familiarity. You need a foundation for

understanding performance, architecture, reliability, and cost trade-offs so you can design, operate, and evolve LLM-based systems with confidence.

What This Book Aims to Do

This book aims to close a critical gap in the GenAI ecosystem: moving from *having* an LLM to *running* LLMs efficiently, reliably, and affordably in real systems.

Our goal is to give you a clear foundation for:

- Understanding what model serving really is and why LLMs fundamentally change the serving problem
- Seeing how LLM execution works (attention, prefill, decode) and how those mechanics shape latency, throughput, and cost
- Building serving systems from scratch so you understand their architecture, caching, and scheduling and the trade-offs behind frameworks
- Measuring performance correctly and making informed engineering decisions instead of guessing
- Applying core optimization techniques—from batching, quantization, and kernel fusion to continuous batching, prefix caching, and speculative decoding
- Choosing and using modern LLM serving frameworks intelligently rather than as black boxes
- Connecting serving to real workloads: chat systems, RAG pipelines, agents, enterprise deployments, and cloud or self-hosted architectures

If you are responsible for making LLM-powered systems actually work—in production, at scale, and within budget—this book is meant to be your practical guide.

Who Should Read This Book

This book is for practitioners who need to move beyond demos and make LLM-powered systems work reliably, efficiently, and at scale. You are likely part of one (or more) of the following groups:

- ML/AI engineers and researchers who have trained or fine-tuned LLM models and now need to serve them efficiently to real users
- Backend and platform engineers who suddenly "own the LLM service," whether on premises, in the cloud, or in hybrid environments
- Data and MLOps engineers who need to extend existing ML platforms to support LLMs, agents, and RAG workloads

- Tech leads and architects responsible for choosing architectures, frameworks, and GPU strategies; and for evaluating the trade-offs between cloud and self-hosting
- Startup founders and small-business builders developing agent platforms or AI products who need to reduce hosting costs, improve reliability, and regain control over performance and economics
- Students and emerging engineers who understand LLM fundamentals and want to learn how real production systems are designed, optimized, and operated

We assume you are comfortable reading Python, are familiar with basic deep-learning concepts, and have at least a passing understanding of transformers and LLMs. You do not need to be a GPU-kernel expert or distributed-systems researcher, but you should be prepared to work with performance metrics, architecture diagrams, and practical system design.

What This Book Isn't

This book is *not*:

- A general introduction to machine learning or deep learning
- A broad "What is GenAI?" or "What can LLMs do?" overview
- A catalog of every LLM product or framework on the market
- A formal survey of research on all possible optimization algorithms

We focus narrowly on serving and optimizing LLMs in real systems. Where background is necessary—for example, to understand concepts like attention, KV cache, or quantization—we explain it just enough to connect it to serving and optimization decisions.

If you're new to machine learning or transformers, we encourage you to pair this book with a more general introduction to deep learning or LLMs, such as *Hands-On Large Learning Models*, by Jay Alammar and Maarten Grootendorst (O'Reilly, 2024), and treat this book as your serving and systems companion.

How This Book Is Organized

This book progresses from foundations, to building systems, to optimization, and finally to frameworks, practical guidance, and future directions.

Chapter 1 introduces model serving and optimization. It explains what models and model serving are, reviews industry practices, and discusses why LLM serving optimization matters.

Chapter 2 focuses specifically on LLM serving, explaining common use cases, execution mechanics (attention, prefill, and decoding), and core serving metrics, supported with code examples.

Chapter 3 teaches you how to design and implement LLM serving systems from scratch, including both single-model and multi-model serving architectures.

Chapter 4 discusses LLM serving best practices, including agent and RAG pipelines, enterprise serving architectures, hosting strategies (buying, self-hosting, or using vendor platforms), and performance measurement.

Chapter 5 explains the core challenges in LLM serving and why they arise from model behavior, hardware constraints, and workload characteristics.

Chapter 6 introduces broadly applicable, essential optimization methods such as continuous batching, quantization, kernel fusion, and prompt prefix caching, supported by practical examples.

Chapter 7 focuses on advanced LLM optimization techniques such as speculative decoding, multi-GPU parallelization, prefill-decode disaggregation, and advanced KV cache management.

Chapter 8 explains why specialized LLM serving frameworks exist and surveys today's leading options, including virtual LLM (vLLM), TensorRT-LLM, SGLang, and llama.cpp, providing guidance on selecting the right framework for your workload.

Chapter 9 walks you through an end-to-end LLM optimization project, helping you build practical intuition for applying optimization techniques to your own use cases.

Chapter 10 looks ahead to emerging directions such as semantic-aware routing, large-scale multi-LoRA serving, multimodal serving, reinforcement learning integration, and edge AI deployment.

How to Use This Book

You can read this book from end to end, but it is also designed for selective use:

- Start with Chapters 1 and 2 if you want a solid foundation in model serving concepts and how LLM serving works.
- Read Chapters 3 and 4 if you want to design, build, and operate real LLM serving systems.
- Use Chapters 5 through 7 if your focus is LLM performance, scalability, and cost optimization.
- Turn to Chapter 8 to understand frameworks and choose the right one.
- See Chapters 9 and 10 for end-to-end applied optimization and future directions.

You can also find all the chapter labs and sample code at the book's GitHub repository.

Wherever you begin, the book provides runnable examples, practical guidance, and lessons from real systems to help you build and improve LLM serving with confidence.

What You'll Need

To get the most out of the hands-on parts of the book, you'll want:

- Access to at least one GPU (cloud or on-premises) capable of running small- to mid-sized LLMs (Google Colab is a good option)
- Familiarity with Python and basic command-line tools
- Comfort with installing and configuring open source serving frameworks, such as vLLM and NVIDIA Triton
- A willingness to experiment: for example, changing batch sizes, model sizes, sequence lengths, and schedules, then measuring the differences

If you don't have direct access to GPUs, you can still benefit from the conceptual and architectural discussions, the results of our pre-executed experiments, and our insight into how trade-offs are made in real LLM serving systems.

Ultimately, our hope is that this book will help you move beyond "we have an LLM endpoint" to "we have a robust, efficient, and understandable LLM serving system"—one that you can reason about, debug, and evolve as the GenAI landscape keeps changing.

We're excited to see what you build.

Conventions Used in This Book

The following typographical conventions are used in this book:

Italic
: Indicates new terms, URLs, email addresses, filenames, and file extensions.

`Constant width`
: Used for program listings, as well as within paragraphs to refer to program elements such as variable or function names, databases, data types, environment variables, statements, and keywords.

This element signifies a tip or suggestion.

This element signifies a general note.

This element indicates a warning or caution.

Using Code Examples

Supplemental material (code examples, exercises, etc.) is available for download at *https://github.com/orca3/llm-model-serving*.

If you have a technical question or a problem using the code examples, please send email to *support@oreilly.com*.

This book is here to help you get your job done. In general, if example code is offered with this book, you may use it in your programs and documentation. You do not need to contact us for permission unless you're reproducing a significant portion of the code. For example, writing a program that uses several chunks of code from this book does not require permission. Selling or distributing examples from O'Reilly books does require permission. Answering a question by citing this book and quoting example code does not require permission. Incorporating a significant amount of example code from this book into your product's documentation does require permission.

We appreciate, but generally do not require, attribution. An attribution usually includes the title, author, publisher, and ISBN. For example: "*Hands-On LLM Serving and Optimization* by Chi Wang and Peiheng Hu (O'Reilly). Copyright 2026 Chi Wang and Peiheng Hu, 979-8-341-62149-7."

If you feel your use of code examples falls outside fair use or the permission given above, feel free to contact us at *permissions@oreilly.com*.

O'Reilly Online Learning

For more than 40 years, O'Reilly Media has provided technology and business training, knowledge, and insight to help companies succeed.

Our unique network of experts and innovators share their knowledge and expertise through books, articles, and our online learning platform. O'Reilly's online learning platform gives you on-demand access to live training courses, in-depth learning paths, interactive coding environments, and a vast collection of text and video from O'Reilly and 200+ other publishers. For more information, visit *https://oreilly.com*.

How to Contact Us

Please address comments and questions concerning this book to the publisher:

O'Reilly Media, Inc.
141 Stony Circle, Suite 195
Santa Rosa, CA 95401
800-889-8969 (in the United States or Canada)
707-827-7019 (international or local)
707-829-0104 (fax)
support@oreilly.com
https://oreilly.com/about/contact.html

We have a web page for this book, where we list errata, examples, and any additional information. You can access this page at *https://oreil.ly/hands-on-llm-serving*.

For news and information about our books and courses, visit *https://oreilly.com*.

Find us on LinkedIn: *https://linkedin.com/company/oreilly*.

Watch us on YouTube: *https://youtube.com/oreillymedia*.

Acknowledgments

Writing a book about a fast-moving field like LLM serving and optimization is both exhilarating and humbling. Many people helped us turn years of experiments, production incidents, and whiteboard sketches into something coherent and useful. We are deeply grateful to everyone who contributed along the way.

First, we would like to thank the team at O'Reilly for believing in this project and for their steady guidance throughout the process—from Nicole Butterfield's support during the proposal stage to our editor Sarah Grey's guidance from early release

through the final manuscript. Your feedback, patience, and high standards not only made this a much better book, but also made the publishing journey smoother and far more enjoyable.

We are also grateful to our employer, Salesforce, for the opportunity to work on advanced agent platforms, which has exposed us to diverse serving patterns and the tremendous challenges of using LLMs in a cost-efficient way. We still recall the many meaningful discussions and proof-of-concept explorations comparing in-house serving solutions, vendor platforms, and hybrid approaches. The chance to build real systems—serving real customers at scale—shaped nearly every chapter in this book. We would especially like to thank Indira Iyer and Bhavesh Doshi for their senior leadership in the AI platform and AgentForce foundation team, and our colleagues for repeatedly delivering meaningful business outcomes in the model serving domain.

Chi would especially like to thank his family for their unwavering support. To my wife, Pei Wu, and our children, Catherine and Tiancheng: thank you for your patience through long evenings and weekends of writing; for reminding me to step away from the keyboard to ski, draw, and play together; and for being the greatest source of motivation I could ask for.

Peiheng would like to thank his family for their constant encouragement and understanding throughout this journey. To my wife, Lillian, and daughter, Iris: your support made it possible to balance work, life, and the many hours spent iterating on code, diagrams, and chapters. As this is my first book, the journey has been a significant learning experience, and I am deeply grateful to have you with me every step of the way. I would also like to thank my parents for their lifelong encouragement and for teaching me the perseverance needed to complete this project.

Finally, to our friends, mentors, reviewers, and the broader open source and research communities around LLMs, serving frameworks, and optimization—thank you for sharing your ideas, tools, and lessons learned openly. This book stands firmly on top of that collective work.

Any mistakes that remain are entirely our own.