

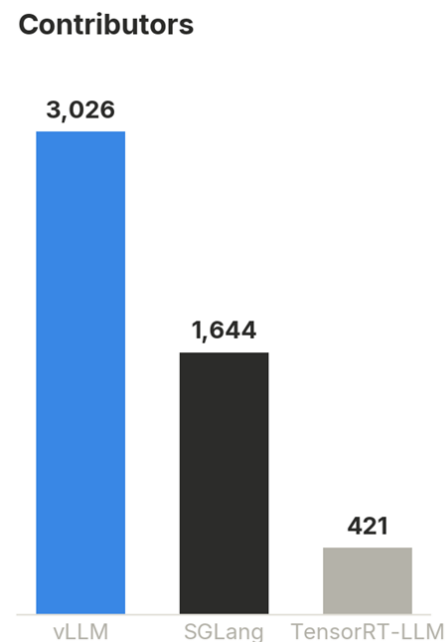
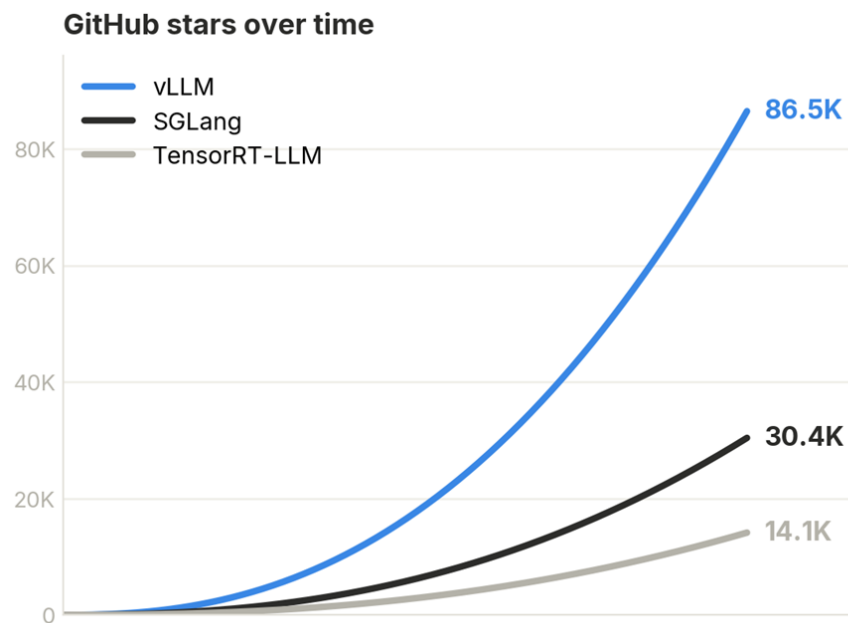
Inside vLLM

Production Best Practices, Model integration and roadmap

vLLM Bangkok Meetup

The state of vLLM today

The open standard for LLM inference and fastest-growing project in the ecosystem



91K

GitHub stars — #1 inference engine

3,336

contributors and counting

https://recipes.vllm.ai/moonshotai/Kimi-K3

--max-model-len

--max-num-seqs



moonshotai/Kimi-K3

Pre-release 2.8T-parameter native multimodal MoE with Kimi Delta Attention, Gated MLA, Attention Residuals, and a 1M-token context window

Pre-release TP8, TEPI6, TP8xPP2, and disaggregated P/D profiles for the 2.8T MXFP4 checkpoint

 View on HuggingFace [↗](#)

 Moe

2.8T / 16 experts/token + shared (of 896 routed)

1,048,576 ctx

vLLM 0.27.0+ 

Multimodal

Text

INSTALL

vLLM 0.27.0+ · CUDA

NIGHTLY

pip / Docker

▼

Verified on NVIDIA GB300 NVL4

HeadNode 1Node 2Node 3

Copy

cURL

Bench

Env

```
docker run --gpus all \
  --privileged --ipc=host -p 8000:8000 \
  -v ~/.cache/huggingface:/root/.cache/huggingface \
  -e GLOO_SOCKET_IFNAME=eth0 \
  -e NCCL_SOCKET_IFNAME=eth0 \
  -e VLLM_ENABLE_K3_LATENT_MOE_TAIL_FUSION=1 \
  -e VLLM_ALLREDUCE_USE_FLASHINFER=1 \
  -e VLLM_ENGINE_READY_TIMEOUT_S=3600 \
  -e VLLM_USE_V2_MODEL_RUNNER=1 \
  -e VLLM_USE_RUST_FRONTEND=1 \
  vllm/vllm-openai:kimi-k3 moonshotai/Kimi-K3 \
  --trust-remote-code \
  --moe-backend auto \
  --gpu-memory-utilization 0.95 \
  --tensor-parallel-size 16 \
  --nnodes 4 \
  --node-rank 0 \
  --master-addr $HEAD_IP \
  --load-format fastsafetensors \
  --no-enable-flashinfer-autotune \
  --max-model-len 1048576 \
  --kv-cache-dtype fp8 \
  --attention-config '{"use_prefill_query_quantization":true}' \
  --enable-prefix-caching \
  --enable-auto-tool-choice \
  --tool-call-parser kimi_k3 \
  --reasoning-parser kimi_k3

# Set $HEAD_IP to the rank-0 node's IP before launch, then run each tab on its own node - # they differ only by --node-rank (TP/TEP) or --data-parallel-start-rank (DEP).
```

HARDWARE

NVIDIA

H100 8×80G

H200 8×141G

B200 8×180G

GB200 NVL4 4×192G

B300 8×268G

GB300 NVL4 4×288G

AMD

MI300X 8×192G

MI325X 8×256G

MI355X 8×288G

VARIANT

MXFP4 1680 GB

STRATEGY

Multi-Node Tensor Parallel

Multi-Node Tensor + Expert Parallel

Multi-Node TP + Pipeline Parallel

Multi-Node Data + Expert Parallel

Multi-Node TP + Data Parallel

Prefill/Decode Disaggregation

Multi-node TP. Splits the model across GPUs spanning multiple nodes. TP is set to total GPU count across all nodes. Head node serves requests, worker nodes run with --headless. Requires InfiniBand interconnect.

Latency oriented

KV OFFLOAD

Off

Simple

Mooncake

LMCache

NODES

Single-node

Multi-node (example: 2) 8×GPU

Multi-node (example: 4) 16×GPU

FEATURES

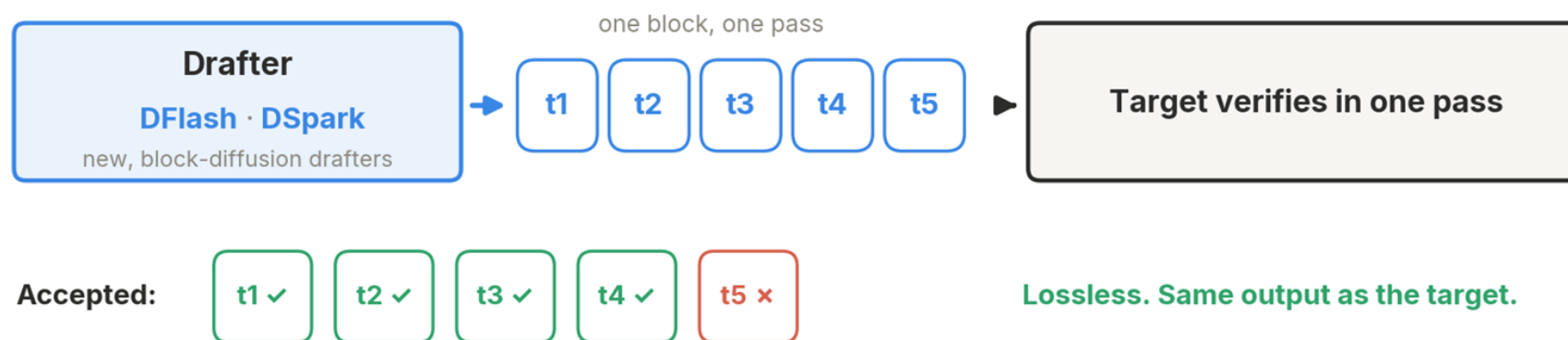
Tool Calling

Reasoning

Spec Decoding (for low latency & small batch size)

Text Only

Speculative decoding



EAGLE-3 has been solid for a while. DFlash and DSpark are the new, fast-moving block-diffusion drafters.

DFlash / DSpark Support

Block-diffusion drafters that draft a whole block in one pass. Example: DSpark speculators offer ~30% higher acceptance length on Qwen3 4/8/14B. Both native in vLLM.

TorchSpec and Speculators

Speculators and TorchSpec train EAGLE-3, DFlash, and DSpark drafters, with unified online and offline training on vLLM hidden states.

Spec Decode as a First-class citizen

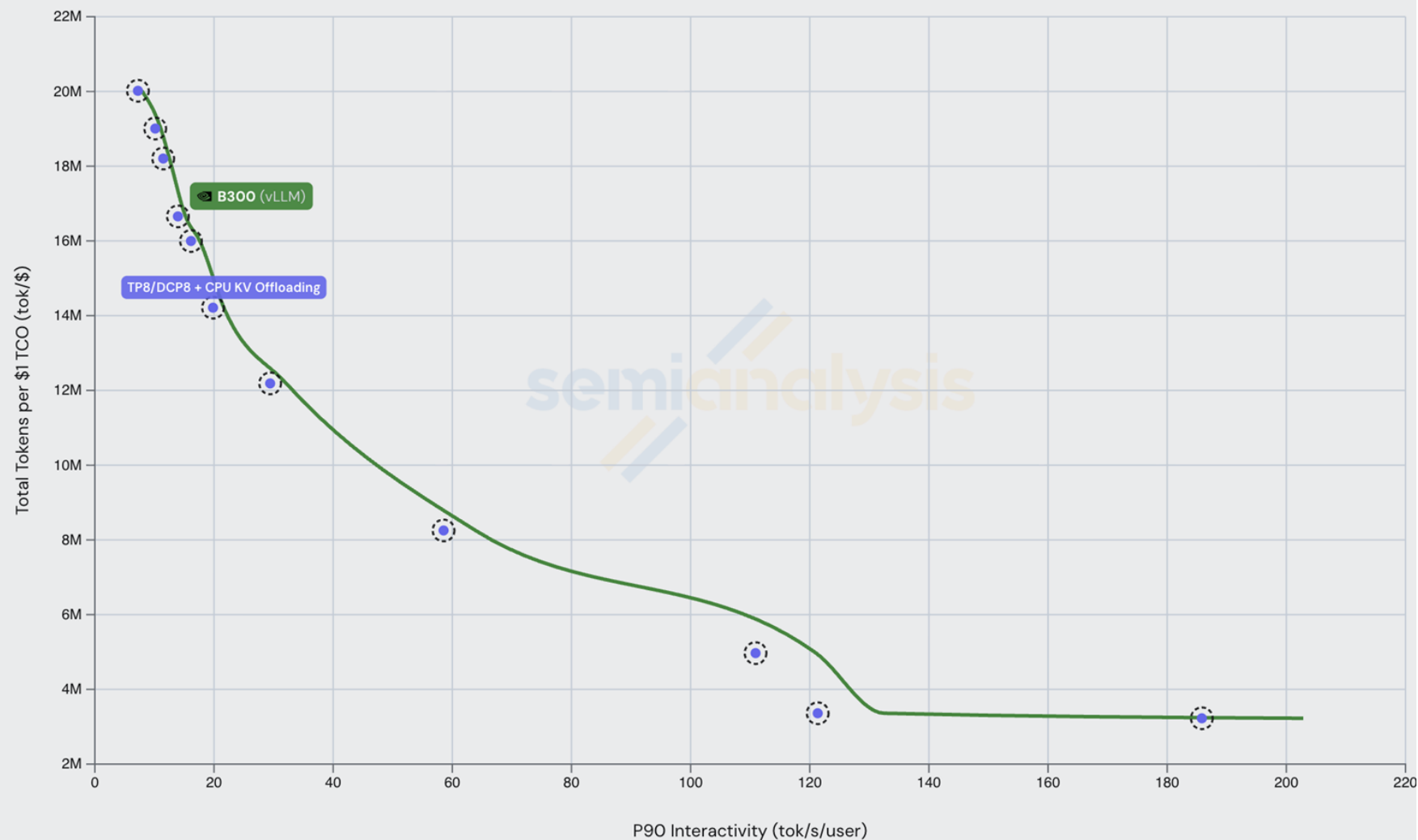
Spec decode works with the rest of vLLM: PD disaggregation, Model Runner V2, prefix caching, tool calling, etc.

Kimi K3 2.8T Agentic Total Tokens per \$1 TCO vs. P90 Interactivity

Cost Tier: Owning Hyperscaler Updated: 2026-09-04 Source: SemiAnalysis InferenceX™

TCO \$/chip/hr: B300: 2.26

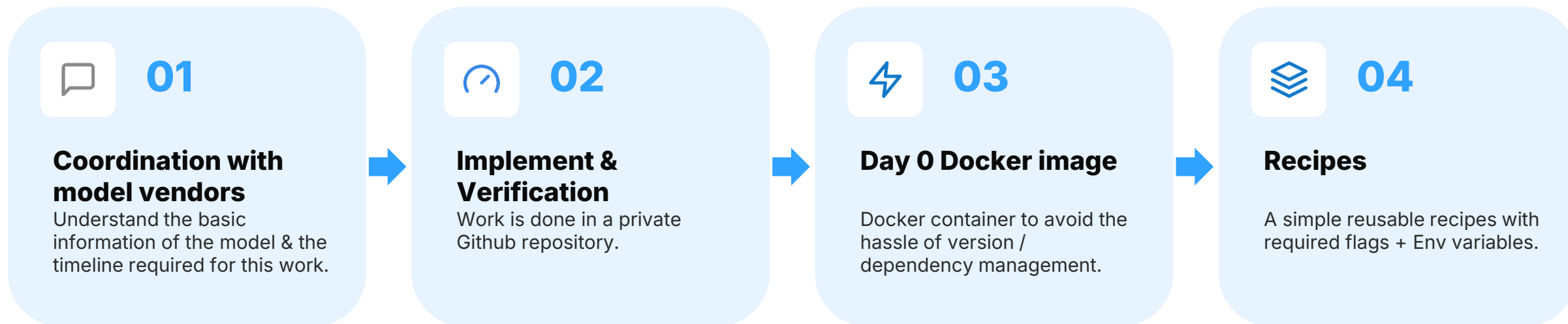
Source: [SemiAnalysis Market July 2026 Pricing Surveys & AI Cloud TCO Model](#)



Shift+Scroll to zoom • Drag to pan • Double-click to reset • Click a point to pin tooltip

The process of adding Day 0 model integration

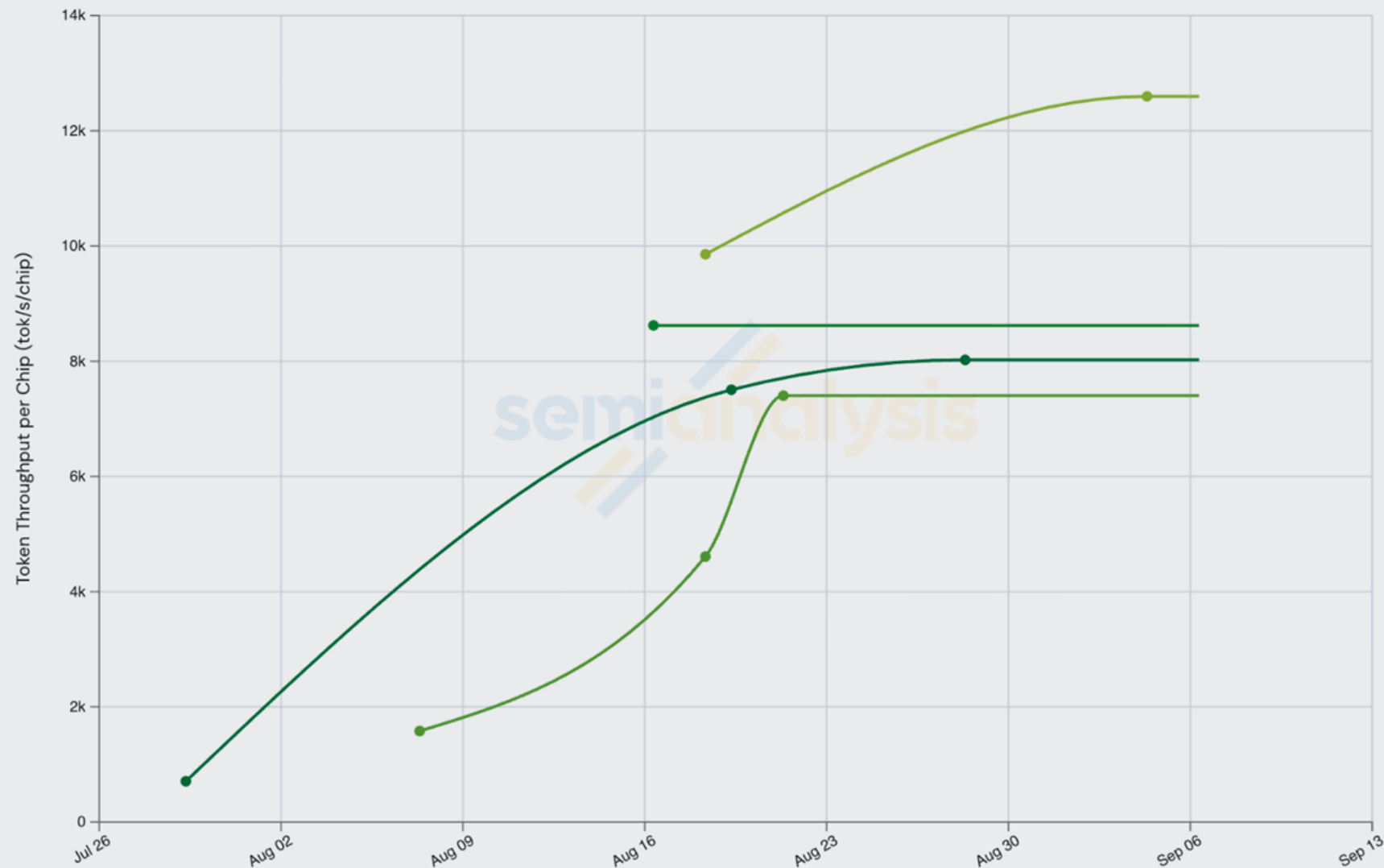
Four stages, using Kimi K3 as a running case study — from initial contact to the integration.



The brutal truth: The whole process could take weeks or even months. And it doesn't stop there.

Token Throughput per Chip Over Time at 35 tok/s/user Interactivity

Model: Kimi K3 2.8T Workload: Agentic Precision: FP4 Metric: Token Throughput per Chip (tok/s/chip) Target: 35 tok/s/user Updated: 2026-09-04 Source: SemiAnalysis InferenceX™



Reset filter



- GB300 NVL72 (Dynamo vLLM)
- GB200 NVL72 (Dynamo vLLM)
- B300 (vLLM)
- B200 (Dynamo vLLM)
- MI355X (ATOM¹)
- MI355X (vLLM)
- H200 (vLLM)

☒ Log Scale

☒ High Contrast

¹ The ATOM engine is promising, however it has yet to serve production tokens. It is still in its infant stage.

Shift+Scroll to zoom horizontally · Drag to pan · Double-click to reset

The Q3 roadmap

Our goal: optimize vLLM as the production agentic serving system.



PRODUCTION SERVING

■ Agentic workloads

Tool calling, structured output, agent-aware caching, and AgentX.

■ Large-scale serving

Dynamo support, disagg, and parallelism across multi-node clusters.



PERFORMANCE

■ Model performance

Day-0 support with leading perf; current sprints are Kimi K3 and GLM 5.2.

■ Speculative decoding

DSpark and DFlash, plus resilience across other frameworks.



REFINEMENT

■ KV cache manager refactoring

Cleaner manager and connector interfaces for offload, reuse, and external stores.

■ CI & release

Cut time-to-signal. Reshape CI to fit how vLLM ships.



ECOSYSTEM

■ Broader hardware support

Optimizations for AMD GPU and Google TPU.

■ RL & Omni

Weight sync, train-inference consistency, and vision models in lockstep with main.



Building the **fastest** and **easiest-to-use**
open-source LLM inference & serving
engine!



<https://github.com/vllm-project/vllm>



<https://docs.vllm.ai>



<https://blog.vllm.ai>



<https://www.zhihu.com/people/vllm-team>



<https://www.linkedin.com/company/vllm-project>



<https://slack.vllm.ai>



https://twitter.com/vllm_project



<https://opencollective.com/vllm>



vllm_project



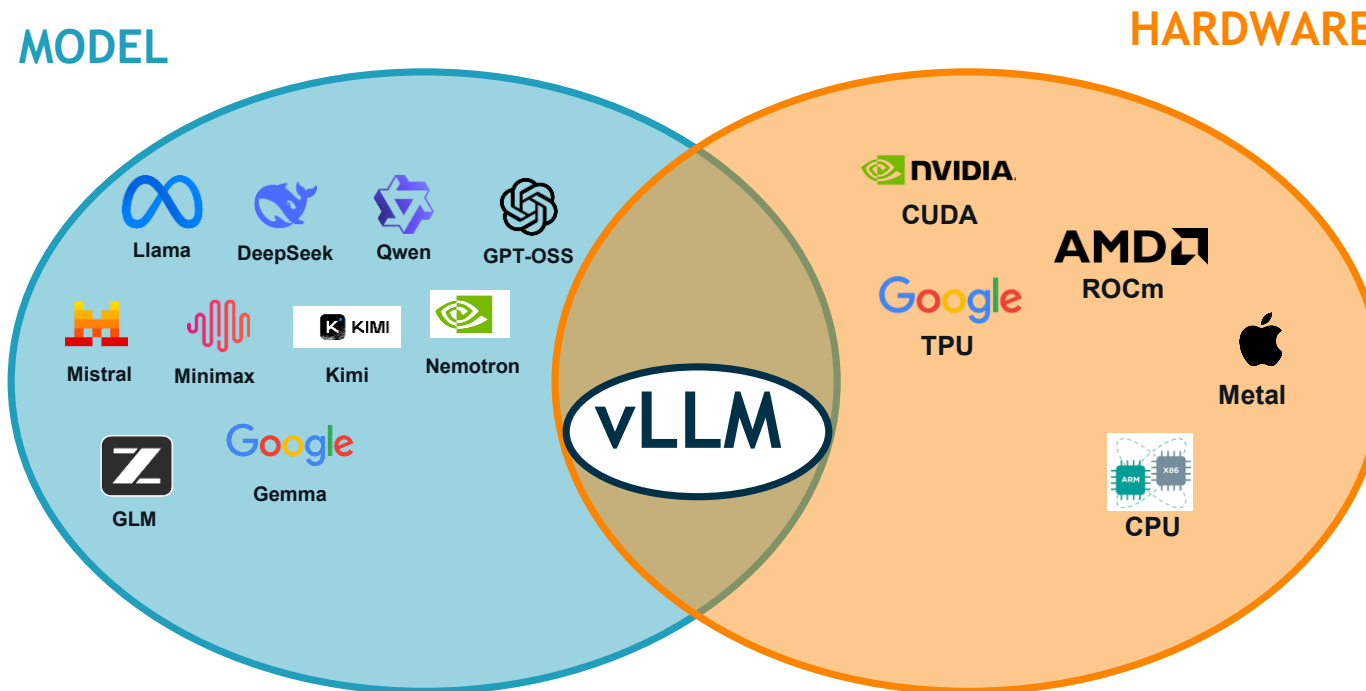
collaboration@vllm.ai

Source: Artificial Analysis · extracted 2026-07-31

[illegible]

What does Inferact do?

Production scaling on data center environment with thousands of GPUs



THANK YOU



Twitter: Xianbao_QIAN

Linkedin: <https://www.linkedin.com/in/tiezhen>

Whatsapp: +852 6206 2575