TURINTECH | ARTEMIS

AI INFERENCE OPTIMIZATION

Cut serving cost and latency on the stack you already run

Day-one ROI | Your existing stack | Code you can audit

Artemis continuously optimizes production AI and software systems. For LLM inference, it improves throughput,latency, and serving cost on your existing stack.

— 01 / THE PROBLEM

Inference cost scales with every user, request, and token. Standard serving stacks — vLLM, TensorRT-LLM, TGI — are built for broad compatibility, not for your specific model, workload, and hardware. Finding better configurations takes repeated experiments and specialist effort most teams can't sustain.

— 02/ WHAT ARTEMIS DOES

Artemis automatically searches the optimization space inside your current serving framework to find faster, lower-cost execution paths that manual tuning is unlikely to uncover.

Every optimization is benchmarked and validated before release, so your platform team gets measurable performance gains while preserving output quality and operational control.

Same model. Same hardware. Same framework. **Just tuned**

REPRESENTATIVE BENCHMARK

● VALIDATED RESULT

THROUGHPUT

+36%

380 tok/s → 517 tok/s

TIME PER OUTPUT TOKEN

-26%

75.3 ms → 55.7 ms

COST PER TOKEN

-26%

throughput-derived, rate-independent

Qwen3-4B AWQ · vLLM · Intel Xeon CPU. Throughput and TPOT measured under concurrent load on vLLM server. Consistent gains observed across model families and hardware classes.

QUALITY ISN'T A PROMISE — IT'S A GATE.

Optimization changes how fast computation happens, not what is computed.

Every Artemis optimization is validated against three independent checks — structural sanity, semantic similarity, and control-prompt validation. If any hard check fails, the optimization doesn't ship.

✓ Structural sanity | ✓ Semantic similarity | ✓ Control-prompt validation

You see faster, cheaper inference. Your customers see no difference.

CONSISTENT GAINS ACROSS THE STACK

Benchmark-validated improvements across CPU, GPU, runtime, and serving layers.

HARDWARE	MODELS / WORKLOAD	RESULT
Intel Xeon CPU	A sweep of 13 models (4B–72B) across the Qwen, Llama, Mistral, and Phi families.	**+8–31%** vLLM throughput
NVIDIA T4 GPU	Whisper speech-to-text	**+25%** runtime acceleration
Intel Arc / Lunar Lake GPU	OpenVINO MVN kernel	**+20%** kernel speedup 8% e2e latency reduction

Don't see your stack? Ask about a custom benchmark on your model and hardware.

WHERE ARTEMIS OPERATES

Most tools fix one layer. Artemis tunes three for compounding gains.

ARTEMIS OPTIMIZATION ZONE

L1	**Serving & Execution** vLLM · SGLang · Triton	ARTEMIS
L2	**Runtime & Compilation** TensorRT-LLM · OpenVINO · ONNX Runtime	ARTEMIS
L3	**Compute kernels** Hardware-specific instruction tuning	ARTEMIS
L4	Hardware / accelerators NVIDIA · AMD · Intel	BASE

Three layers.
One optimization loop.

- Serving configuration
- Runtime behavior
- Hardware-specific kernels

Every result is validated before release.

WHERE ARTEMIS PAYS OFF MOST

If inference is a meaningful COGS line in your AI product

Vertical SaaS · copilots · agentic products

Every point of unit-cost reduction flows directly to gross margin. Latency reductions improve product experience and reduce churn risk.

If serving efficiency is your competitive moat

Inference platforms · internal AI platforms · model-serving infra

Better tok/s per dollar *is* the product. Defend price-performance against new entrants without expanding your perf-engineering team.

Get a custom benchmark on your stack

See your day-one ROI before you commit. Real before/after numbers on your model and hardware.

sales@turintech.ai

turintech.ai

ARTEMIS ALSO OPTIMIZES — Agentic Software · Latency-Critical Infrastructure · Planning & Scheduling

© 2026 TurinTech AI

Results vary by codebase and workflow.