Angel Marinov.

AI SYSTEMS & PERFORMANCE ENGINEER



Performance, measured.
Engineering, owned.

I build and optimize the parts of AI systems where execution cost accumulates: inference paths, GPU kernels, retrieval, and runtime internals. My work connects a focused code change to a clearly scoped, measured result.

4.38×

Persistent KV reuse

Batch-time speedup versus prefix recomputation.
NVIDIA A40 · 16K · C=8.

38.1%

Lower matmul latency

Best tested kernel configuration.
M4 Max · proposed upstream patch.

72×

Wavefunction grid speedup

Against a vectorized NumPy path.
Single thread · one orbital.

Runtime & serving

Inference paths, cache lifecycles, memory traffic, and workload-specific performance.

Kernels & retrieval

Compiled numerical routines, GPU dispatch, lexical ranking, and filtered search.

Bring one real bottleneck.

www.0xamlab.com/contact.html

github.com/0xamlab linkedin.com/in/aimarinov

START A PROJECT

Less repeated work.
More useful computation.

Four workload-specific results across persistent inference caching, GPU dispatch, concurrent model serving, and compiled scientific computation.

NVIDIA / SERVING

Published benchmark

Persistent KV reuse

4.38×

batch wall-time speedup

A40 · Qwen2.5-7B · BF16 · 16K prefix · concurrency 8

23.66 s to 5.40 s versus forced prefix recomputation. Same-process native-hot caching was faster at 2.42 s. Three repetitions per cell.

Built the external cache connector and its benchmark / lifecycle validation.

github.com/thyn-ai/telys-vllm

GPU / QUANTIZED MATMUL

Open upstream PR

Quantized matrix multiplication

38.1%

lower latency in the best tested cell

M4 Max · 4-bit weights · BF16 activations · M=2048

Best of five tested configurations; reported range 5.6–38.1% lower latency. A kernel result, not an end-to-end model speedup. Submitted PR.

Authored the proposed tile-dispatch optimization and regression coverage.

github.com/ml-explore/mlx · PR 4826

NVIDIA H100 / SERVING

Archived result

27B-model concurrent serving

781.3 tok/s

aggregate output throughput

H100 80GB · Gemma 3 27B · FP16 · 550 in / 256 out

One H100 at concurrency 50. Recorded in the August 26, 2026 documentation snapshot; not a newly rerun comparison.

Built and benchmarked the serving runtime described in the source documentation.

docs.algenta.ai · NVIDIA serving

MOJO / SCIENTIFIC COMPUTE

Published benchmark

Molecular wavefunction grids

72×

speedup versus the vectorized baseline

M4 Max · single thread · benzene / 6-31G* · 50^3 grid

689 ms to 9.5 ms versus the vectorized pyquante2 / NumPy path. A single-orbital grid, not a complete chemistry simulation.

Built the native Gaussian-grid evaluation path and Python integration.

github.com/thyn-ai/mojo-kernels

Evidence snapshot: November 9, 2026. Results are author-published, not independently certified or newly rerun for this brochure. “Open upstream PR” describes the reviewed snapshot; submission does not imply upstream adoption.

Find the cost.
Change the mechanism.

Compiled search, bounded memory, physical partitioning, and resident execution. Each result keeps its measurement boundary.

38.7×

warm-query speedup in the selected cell

Published benchmark

Fuzzy search at 100K documents

M4 Max · 100K docs · 16-code-unit patterns · warm

1,239.508 ms to 32.003 ms versus Fuse.js 7.1.0. Accelerated path uses 16 threads; warm-query measurement.

github.com/thyn-ai/mojo-kernels

1.10–1.95×

warm-query speedup across the tested grid

Published benchmark

BM25 versus an optimized scorer

M4 Max · 1K–100K docs · 5–20 query terms

Versus the best measured bm25s / Numba variant across nine cells. Numerical parity is against rank_bm25, not bm25s.

github.com/thyn-ai/mojo-kernels

4,099 QPS

at concurrency 8; recall reported as 1.0000

Open upstream PR

Filtered vector retrieval

Cohere-100K · 768 dimensions · cosine · 1% filter

Concurrency 8; reported recall 1.0000. The separate 1.5 ms serial p99 is not a concurrency-8 latency. Hardware unspecified in PR summary.

github.com/zilliztech/VectorDBBench · PR 824

0.83–0.93 ms

documented warm finance.npv execution

Documented observation

Warm deterministic execution

One representative operation · warm resident runtime

One documented warm finance.npv call; not an agent workflow or a matched benchmark campaign. First-time provisioning excluded.

github.com/thyn-ai/sqai-mcp · documentation

19.51 MB

peak RSS in the selected scored test

Open upstream PR

Bounded phrase-query memory

Repeated-token phrase stress case · scored search

Selected repeated-token phrase stress test. Earlier implementation crossed a 256 MiB watchdog; its exact peak was not measured. Submitted PR.

github.com/quickwit-oss/tantivy · PR 3248

A defined problem. A reviewable result.

Discuss your project at 0xamlab.com