

Math Unikernel

Benchmark Results — Bare Metal vs. Linux
2048×2048 GEMM & Dot Product | AVX/FMA | x86-64

Executive Summary

Math Unikernel is a bare metal x86-64 unikernel OS designed for high-performance compute workloads. By eliminating the OS scheduler, virtual memory overhead, and system call boundaries, it allows AVX/FMA workloads to execute with direct hardware access. These benchmarks measure single-precision floating point performance against an equivalent Linux implementation on identical hardware.

Metric	Result
GEMM Speedup (Unikernel vs Linux)	1.35x (35.2% faster)
Dot Product Speedup	1.01x (1.4% faster)
Overall System Speedup (Amdahl's Law)	1.31x (30.6% faster)
GEMM Throughput — Unikernel	15.92 GFLOPS
GEMM Throughput — Linux	11.78 GFLOPS

GEMM — 2048×2048 Single Precision

General Matrix Multiplication (GEMM) is the dominant operation in deep learning inference, accounting for the majority of floating point work in transformer and CNN models. A 2048×2048 GEMM requires 17.18 billion floating point operations per call. At this size the working set (three 16MB matrices = 48MB) exceeds L3 cache on most processors, making the benchmark representative of real-world inference workloads.

Platform	Avg Time (ms)	Avg Cycles	GFLOPS	Iterations
Unikernel (Bare Metal)	1079	3,135,425,891	15.92	10
Linux	1459	4,238,388,002	11.78	10
Speedup	1.352x	1.352x	1.352x	—

The unikernel completed GEMM in 1079ms on average versus 1459ms on Linux — a 35.2% reduction in execution time. This translates to 15.92 GFLOPS on bare metal compared to 11.78 GFLOPS on Linux.

Dot Product — 524,288 Elements

The dot product benchmark measures vectorized inner product performance over 524,288 single-precision floats (2MB per vector), using AVX/FMA instructions. At this size both vectors fit comfortably in L3 cache, making this primarily a compute-bound test rather than a memory bandwidth test.

Platform	Avg Time (ms)	Avg Cycles	MFLOPS	Iterations
Unikernel (Bare Metal)	9.00	26,420,990	116.5	100
Linux	9.13	26,503,965	114.8	100
Speedup	1.014x	1.003x	1.014x	—

Dot product results are nearly identical between platforms (9.00ms vs 9.13ms), indicating that for compute-bound workloads that fit in cache, the OS scheduling overhead is minimal. The significant GEMM advantage demonstrates that the unikernel's benefit emerges most clearly in memory-intensive workloads where cache thrashing and scheduler preemption have a compounding effect.

Amdahl's Law Analysis

Applying Amdahl's Law to estimate overall inference speedup, assuming GEMM accounts for 90% of total inference compute (a conservative estimate for transformer models where attention and FFN layers are GEMM-dominated):

$$S = 1 / ((1 - 0.9) + 0.9 / 1.352)$$

$$S = 1 / (0.10 + 0.6656)$$

$$S = 1.306x \text{ (30.6\% overall speedup)}$$

A 30.6% overall system speedup means the same hardware can serve 30.6% more inference requests per second, or equivalently, the same throughput can be achieved with fewer servers. At cloud scale this compounds significantly across millions of daily inference calls.

Methodology

Parameter	Value
Hardware	Panasonic CF-33 Toughbook, x86-64

GEMM Matrix Size	2048 × 2048 single-precision float
GEMM Working Set	3 matrices × 16MB = 48MB
Dot Product Length	524,288 elements (2MB per vector)
SIMD Instructions	AVX + FMA (256-bit YMM registers)
Timing Method	RDTC (CPU cycle counter)
GEMM Iterations	10
Dot Product Iterations	100
Unikernel Compiler	GCC -O2 -march=native -ffreestanding
Linux Comparison	Equivalent C implementation, same compiler flags