Cachee Achieves 28.9-Nanosecond Cache Reads - Verified as Fastest Full-Featured Cache Engine Ever Benchmarked

At 100 billion lookups/year, a server tied to Elasticache would spend more than 390 days of time in wasted cache time. Cachee reduces that to 48 minutes.

Cachee – The Intelligent Caching Platform

RIVERVIEW, FL, UNITED STATES, April 3, 2026 /EINPresswire.com/ -- Cachee, Inc. today published independently reproducible benchmark results demonstrating 28.9-nanosecond GET latency for its in-process cache engine — the lowest latency ever measured for a cache with eviction intelligence, TTL management, and enterprise features. The benchmark suite, which runs on any modern computer, shows Cachee adds only 3.6 nanoseconds of overhead versus a raw concurrent hash map, while delivering 46 enterprise capabilities including the industry's first post-quantum cryptographic cache attestation.

Verified, Reproducible Results

The benchmark suite, published as open-source at cachee.ai/benchmark, measures three cache backends on identical hardware, identical workloads, and identical methodology:

Backend | GET Latency (100K keys, 256B values) | What It Is |

Raw HashMap (Mutex) | 22.8 ns | Theoretical minimum. No concurrency safety. |
DashMap (lock-free) | 25.2 ns | Industry-standard concurrent map. No eviction. |
Cachee L0 | **28.9 ns** | Full cache engine: eviction, TTL, frequency tracking, 46 features. |

Each data point represents 10 measured iterations after 3 warm-up runs, with full statistical

> Everyone pays for faster internet. For everyone who touches your product, a faster cache is a faster internet. 11,726 times faster”
>
> *Eric Beans - CEO - H33.ai, Inc.*

reporting including median, P95, P99, and standard deviation. The benchmark SHA-256 verification hash is `77a9d4f0e5696b864779821db8495229b23e762dd1049f994267fadf933e6c7a`.

"The overhead of a complete enterprise cache engine should be measured in nanoseconds, not milliseconds," said Eric Beans, CEO of Cachee, Inc. "At 3.6 nanoseconds of overhead, Cachee proves that intelligent caching and raw speed are not a tradeoff."

How 28.9 Nanoseconds Compares

Cache | GET Latency | Cachee Advantage |

Cachee L0	**28.9 ns**	—
Moka (Rust, W-TinyLFU)	~50 ns	1.7x faster
Caffeine (Java, W-TinyLFU)	~65 ns	2.2x faster
Redis (localhost)	~500 ns	17x faster
AWS ElastiCache (same-AZ)	~339,000 ns	11,726x faster

First Cache with Post-Quantum Cryptography

Cachee v4.3 includes ML-DSA-65 (Dilithium) attestation signatures on cache entries - the first implementation of NIST FIPS 204 post-quantum cryptography in any caching product. This enables tamper detection and authenticity verification for cached data, secured against both classical and quantum computing attacks.

Additionally, Cachee implements ML-KEM-768 (Kyber) for post-quantum key exchange during cache replication, ensuring data in transit is protected against harvest-now-decrypt-later attacks.

46 Enterprise Capabilities in a Single Binary

The 28.9-nanosecond latency is achieved while delivering:

- 4 Protocol Interfaces**: REST, RESP (177+ Redis commands), gRPC, and QUIC with 0-RTT
- Advanced Cache Intelligence**: CDC auto-invalidation via Postgres WAL, Causal Dependency Graphs with DAG cascade invalidation, Cache Contracts for SLA enforcement, Semantic Invalidation, Speculative Pre-Fetch, and Cost-Aware Eviction
- Data Consistency**: MVCC with zero-lock readers, Temporal Versioning for time-travel reads, Cross-Cluster Coherence protocol

- Security**: AES-256-GCM encryption at rest, RBAC with role-based access control, tamper-evident audit logging
- Infrastructure**: Raft consensus, disaster recovery with signed snapshots, auto-scaling, multi-tenant namespace isolation, Lua scripting

Reproduce the Benchmark

The complete benchmark suite is available in the Cachee repository:

```
cd rust && cargo run --release --example benchmark_suite
```

The suite runs on any x86_64 or ARM64 system with Rust installed. No special hardware required. Results vary by CPU architecture but the relative performance between backends remains consistent.

About Cachee

Cachee, an [H33.ai, Inc.](https://h33.ai) company, is an enterprise-grade in-process cache engine built in Rust. It deploys as a sidecar container or embedded library, delivering nanosecond-speed reads with zero network hops. Cachee is compatible with AWS ElastiCache, Redis, DynamoDB, Memcached, and CloudFlare KV as L2 backing stores. Cachee is available on the AWS Marketplace and at cachee.ai.

Media Contact

Eric Beans, CEO
Cachee, Inc.
press@cachee.ai
cachee.ai

Benchmark methodology: 100,000 pre-allocated keys, 256-byte values, single-threaded sequential GET, 3 warm-up + 10 measured iterations, release build with LTO. Full statistical reporting available in benchmark output. Comparison numbers for Moka, Caffeine, Redis, and ElastiCache sourced from published maintainer benchmarks and documented production measurements. Apple M4 Max, 16 cores, 128 GB RAM.

Keywords

Fastest cache, nanosecond cache, cache benchmark, Cachee, in-process cache, Rust cache

engine, cache latency, Redis alternative, ElastiCache alternative, post-quantum cache, ML-DSA-65, Dilithium cache attestation, cache eviction algorithm, Cachee-FLU, W-TinyLFU alternative, low latency caching, sub-microsecond cache, cache acceleration, L1 cache, distributed cache, cache optimization, 28.9 nanoseconds, 11726x faster, reproducible benchmark, enterprise cache, cache comparison 2026, DashMap, Moka, Caffeine, Raft consensus cache, MVCC cache, CDC auto-invalidation, vector search cache, QUIC cache protocol, gRPC cache, RESP protocol, 177 Redis commands, zero-trust cache, encrypted cache, AES-256 cache, AWS Marketplace cache, sidecar cache, database acceleration, cache hit rate, 99 percent hit rate

Patent Pending: U.S. Application No. 19/309,560

Eric D Beans
H33.ai, Inc.
+1 813-464-0945
email us here
