

The 4x Working Set Multiplier: Defeating the Memory Wall Without Buying More Silicon

52 GB owned. 174 GB effective. 0 GB rented.

Effective means the managed-heap DRAM that the same information would need if it were held as ordinary V8 objects.

Author: Brennan DeCrow

Affiliation: ATESO Labs / ManyMoats Research

Parent Research Authority: *ATESO & The Magma Runtime: The Thermodynamic Obsolescence of Document-Centric Execution for Continuous Spatial Compute* (DeCrow, 2026; USPTO App #64/159,586) [1]. Formal treatment: DeCrow (2026b), preprint v2 [2].

Status: M-Tier Master White Paper. Gold Master Candidate (post-Stage 5 synthesis, 22 September 2026)

Review lineage: DeepSeek (physics & arithmetic) → Muse (spatial & typography) → GLM (distributed systems) → Kimi/V4 (long-context consistency) → Claude Opus 5.5 (synthesis & certification)

Next stage: Headless Fable & Astra final stamp

Evidence notation (inherited from the parent paper). **F** formal result under stated assumptions · **R** reproduced from an executable artifact · **U** author-reported, raw data not in the package · **V** independently checked against a vendor, standards or market-data source · **P** proposed design or experiment not yet run. Appendix B assigns a mark to every number in this paper.

Abstract

In memory-heavy servers, DRAM is now the largest line on the invoice. Yet much of the DRAM a managed runtime occupies holds no information.

We measured it. A 64-byte numeric record held as an ordinary V8 object occupies 224 bytes of live heap ($\beta = 3.50$) before any collector headroom. Across nineteen representations in three runtimes, the same record costs between 64 and 580 bytes.

ATESO stores the record as a 64-byte granule plus 8.06 bytes of directory and Merkle metadata. It carries no object headers and needs no collector.

The result is a closed-form law. When the representation being replaced has $\beta = 3.5$, every resident ATESO byte carries the state of 4.14 bytes of managed heap (Sovereign Mode). The figure falls to 2.20 bytes when unpartitioned writers force 128-byte padding (Siege Mode).

On a 52 GB two-Mac reference cluster with a 10 GB operating reserve, the law yields 174 GB of managed-heap equivalent; on a 256 GB server, about 1 TB. We also state where it stops: against a flat, off-heap buffer, ATESO is 11% larger, not smaller.

The Law of Working Set Multipliers

Let a workload hold S bytes of information. A managed runtime needs M_{heap} bytes of DRAM to hold it; ATESO needs M_{ATESO} . The working-set multiplier is their ratio:

$$W = \frac{M_{\text{heap}}}{M_{\text{ATESO}}} \geq \beta (1 - \alpha_{\text{GC}})(\gamma + f)$$

Symbol	Meaning	Value used
β	Live managed bytes per byte of information	3.5 (measured, Appendix A)
α_{GC}	Fraction of the heap a collector keeps free	0.25 (conservative, §2.2)
γ	Slots per granule: 1 packed, 2 padded to a 128-byte line	1 or 2
f	ATESO metadata per payload byte	0.12598, budgeted at 0.126

Like Amdahl’s law [3], it is exact given its inputs. Its empirical content lives in β , which Appendix A measures rather than assumes.

SOVEREIGN MODE	$W = 4.14\times$	one writer per 128-byte line, 64-byte granules ($\gamma = 1$)
packed		
1)		
equivalent	42 GB arena	→ 174.1 GB managed-heap
effective)	(52 GB cluster	→ 3.35× system-level
SIEGE MODE	$W = 2.20\times$	unpartitioned writers, granules padded
fallback		to a full 128-byte line ($\gamma = 2$)
2)		
	42 GB arena	→ 92.2 GB managed-heap

equivalent |

$$\beta = 3.5 \text{ measured (V8 objects)} \cdot \alpha_{GC} = 0.25 \cdot f = 0.126 \cdot 10 \text{ GB reserve}$$

1. The CFO Invoice: Why Enterprise Memory Is Broken

1.1 What memory costs now

- Conventional DRAM contract prices rose 93–98% quarter on quarter in Q1 2026, and DRAM industry revenue rose 81% to \$97 billion [T7]. TrendForce had projected server DRAM alone at about +90%, the largest quarterly increase it has recorded [T1].
- A 64 GB DDR5 RDIMM carried a Q1 2026 contract price of \$820–950, up 80–120% on the prior quarter [T2]. Citi projects \$1,586 by Q4 2026 [T3].
- For scale: Samsung’s 32 GB DDR5 module sold for \$149 in September 2025 and \$239 two months later [T4].
- The climb has slowed, not reversed. TrendForce expects server DRAM contract prices to rise a further 13–18% in Q3 2026 [T5].
- In a memory-heavy dual-socket reference build (512 GB DDR5), memory’s share of the bill of materials rose from about 18% in January 2025 to about 53% in July 2026 [T6].

At those prices, the 256 GB in the reference server below costs \$3,280–3,800 in Q1 2026 contract terms, and \$6,344 at Citi’s Q4 2026 projection.

1.2 The invoice

	Server + managed runtime	Server + ATESO	Two Macs + ATESO
Physical DRAM	256 GB DDR5 RDIMM	256 GB DDR5 RDIMM	52 GB unified (36 + 16)
Reserved for OS and runtime	8 GB	8 GB	10 GB (6 + 4)
State-bearing DRAM	248 GB	248 GB	42 GB
Information held	53.1 GB	220.3 GB	37.3 GB
Managed-heap equivalent	248.0 GB	1,027.8 GB	174.1 GB
Hardware price	\$15,200	\$15,200	\$5,598 (list)
	\$286	\$69	\$150

	Server + managed runtime	Server + ATESO	Two Macs + ATESO
Price per GB of information			

- **Same silicon, 4.14× the information.** Moving the workload from managed objects to resident granules on the same server cuts the price per GB of held information from \$286 to \$69. This is the paper’s central claim. It involves no change of hardware class and no change of price.
- **Owned laptops against a rack server.** The two Macs hold 70% of the server’s information at 37% of its price and 20% of its DRAM. That is 1.9× lower cost per GB of information.
- **What the table does not claim.** It is not a like-for-like reliability comparison. Laptops lack registered ECC DIMMs, out-of-band management and rack-grade redundancy. The table prices capacity, not availability.

Mac prices are Apple list prices: \$3,599 for the 14-inch M5 Max (36 GB, 2 TB) [A1] and \$1,999 at launch for the 14-inch M1 Pro (16 GB) [A6]. The server figure is the author’s reference quote for a dual-socket 2U configuration.

[Fig. 1 – DRAM needed to hold the cluster's 37.3 GB of information]

Managed heap: V8 objects ($\beta = 3.5$) + 25% collector headroom
174.1 GB

└ information	37.3 GB
└ headers, tagged slots, boxed numbers	93.3 GB
└ collector headroom	43.5 GB

ATESO resident arena
42.0 GB

└ information	37.3 GB
└ directory + Merkle tree	4.7 GB

2. The Root Cause: Managed Heaps Carry Air

2.1 Where the bytes go (measured)

We held one million copies of one record in each representation. The record is eight float64 values, 64 bytes of information. We then measured live heap after a full collection (Appendix A).

- **V8 (Node.js 22).** An object with eight numeric fields is 88 bytes: a 24-byte header (hidden-class map, property and element pointers) and eight 8-byte tagged slots. Each non-integer value lives in its own 16-byte heap-number box. With the 8-byte array slot that references it, the record costs 224 bytes, so $\beta = 3.50$ exactly. Object literals, class instances and `JSON.parse` output all measure the same.

- **V8 representation is global.** V8 records a field’s representation on the object’s shape. Eight small integers cost 96 bytes ($\beta = 1.50$) in a fresh shape. They cost 223 bytes once any object of that shape has stored a fractional value. One stray double can box a field for every object of that shape.
- **JVM (HotSpot 21).** Primitive fields are stored inline. A class with eight double fields costs 84 bytes ($\beta = 1.31$). Boxing the fields as `Double` raises that to 244 bytes ($\beta = 3.81$). A `HashMap<String, Double>` per record costs 580 bytes ($\beta = 9.06$).
- **CPython 3.11.** Every float is a separate 24-byte object. A `__slots__` class costs 296 bytes ($\beta = 4.63$), a dataclass 344 bytes ($\beta = 5.38$) and a dict 472 bytes ($\beta = 7.38$).
- **Packed buffers.** `Float64Array`, `double[]`, `array('d')` and NumPy cost 64–65 bytes ($\beta \approx 1.0$).

β is a property of the representation, not of the language. The reference value, $\beta = 3.5$, is the measured cost of an ordinary JavaScript object in Node.js. The Node.js build we measured has V8 pointer compression turned off (`v8_enable_pointer_compression = 0`), so tagged slots are 8 bytes wide. Builds that enable compression [8], as Chrome does, use 4-byte slots and have a smaller β .

2.2 The collector’s tax

A tracing collector needs free space to work. We charge it $\alpha_{GC} = 0.25$, a heap $1.33\times$ the live data.

This is generous to the managed runtime. Hertz and Berger found that a generational collector matched explicit memory management only when given five times as much memory. With three times as much, it ran 17% slower; with twice as much, nearly 70% slower [6]. Any larger headroom raises W , so on this term the paper’s figure is conservative.

2.3 The expansion identity

$$M_{\text{heap}} \geq \beta S (1 - \alpha_{GC}) M_{\text{heap}} \geq \frac{\beta}{1 - \alpha_{GC}} S$$

For illustration, 40 GB of information at β between 3.2 and 4.1 needs 170.7–218.7 GB of managed heap. Across the measured range, β from 1.0 to 9.06, the same 40 GB needs 53–483 GB.

2.4 What ATESO keeps

Every record is a 64-byte granule in a preallocated, cache-aligned arena (`.many`). Its metadata is fixed:

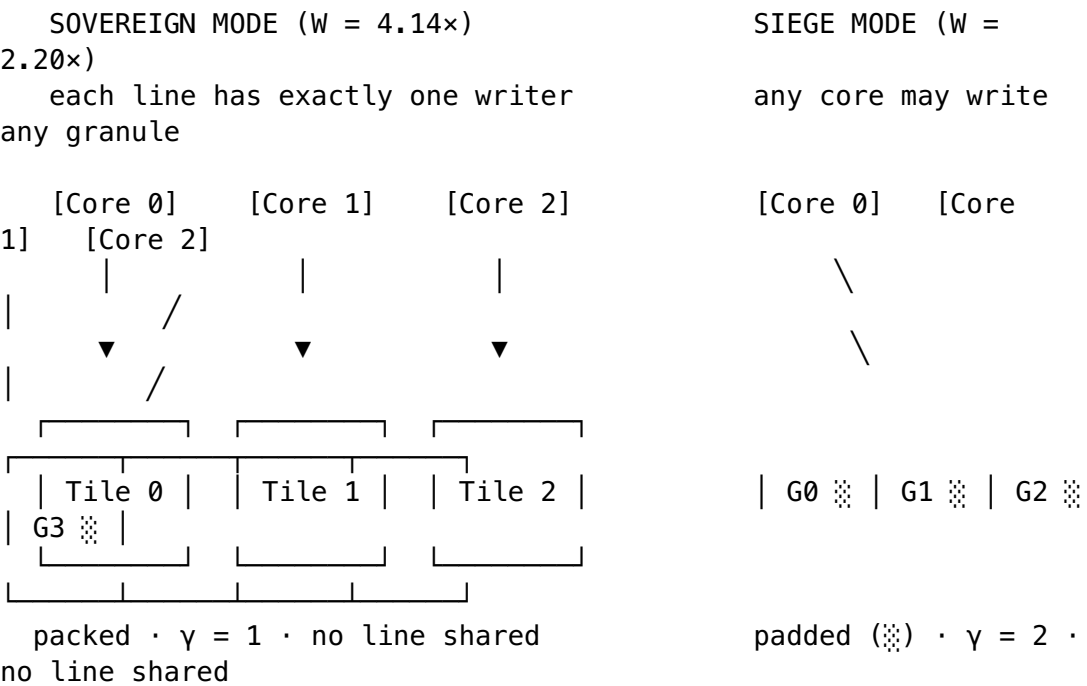
Component	Bytes per granule	Share of payload
Payload	64	—
Directory entry	8	12.5%
Merkle tree (64 KiB leaves, 32-byte digests)	0.0625	0.098%

Component	Bytes per granule	Share of payload
Total	72.0625	$f = 12.598\%$

The arena has no object headers, per-field pointers, boxed numbers or collector headroom. The overhead is not zero: it is 12.6%, fixed and paid once.

3. The Two Regimes: Sovereign Mode vs. Siege Mode

[Fig. 2 – Regime topology]



3.1 Closed-form derivation

The managed heap is bounded below. ATESO’s metadata is bounded above:

$$M_{\text{heap}} \geq \beta(1 - \alpha)GC = 3.50.75 = 4.6667M_{\text{ATESO}} = \gamma + f \leq \gamma + 0.126$$

$$\frac{M_{\text{heap}}}{M_{\text{ATESO}}} \geq \frac{\beta(1 - \alpha)GC}{M_{\text{ATESO}}} = \frac{3.5}{0.75} = 4.6667$$

$$\frac{M_{\text{heap}}}{M_{\text{ATESO}}} \leq \frac{\gamma + f}{M_{\text{ATESO}}} = \frac{\gamma + 0.126}{M_{\text{ATESO}}}$$

The multiplier is therefore bounded below:

$$W \geq 4.6667\gamma + 0.126$$

3.2 Sovereign Mode: $W = 4.14\times$

- **Discipline.** Every 128-byte line has exactly one writer per epoch. State flows over single-producer/single-consumer rings (*.spine*). CPU workers own disjoint, line-aligned tiles. On the GPU, XPBD constraint projection runs over

graph-colored partitions, so no two constraints of one color write the same particle [11].

- **Stride.** $\gamma = 1$. Granules pack two to a 128-byte line.
- **Result.**

$$W_{\text{sovereign}} \geq 4.6667 \cdot 1.126 = 4.1445 \quad W_{\text{sovereign}} \geq \frac{4.6667}{1.126} \approx 4.1445$$

3.3 Siege Mode: $W = 2.20\times$

- **When it applies.** Siege Mode covers unpartitioned multi-writer access on hardware whose coherence granule is 128 bytes. Apple Silicon reports a 128-byte line (`hw.cachelinesize`) [A2]. On the M1 generation, independent measurement puts the P-core L1D at 64-byte lines and the shared L2 at 128-byte lines [A3]. Two 64-byte granules therefore share one coherence granule.
- **The choice.** Sharing a line does not change footprint; it changes throughput. When two cores write the same line, it bounces between them (false sharing) [9]. The runtime can pay in time: keep granules packed ($W = 4.14\times$) and absorb the coherence traffic. Or it can pay in space: pad every granule to a full line, which sets $\gamma = 2$:

$$W_{\text{siege}} \geq 4.6667 \cdot 2.126 = 2.1950 \quad W_{\text{siege}} \geq \frac{4.6667}{2.126} \approx 2.1950$$

- **Scope.** On x86-64 and Neoverse-class servers the line is 64 bytes. A 64-byte granule already owns its line there, so Siege padding is unnecessary.

[Fig. 3 – One 128-byte coherence granule, three layouts]

0B	64B	128B
Sovereign: packed, Contended: packed, padded,	Granule A (Core 0)	Granule B (Core 0)
	one writer, no bounce	
	Granule A (Core 0)	Granule B (Core 1)
bounces	line	
	Granule A (any core)	padding
	Siege: no bounce, 2× space	

3.4 Sensitivity to the representation being replaced

Representation replaced (measured)	β	W Sovereign	W Siege	Managed heap for the cluster's 37.3 GB*
Packed buffer on a collected heap (double[])	1.00	1.18×	0.63×	49.7 GB
JVM class, primitive doubles	1.31	1.55×	0.82×	65.3 GB
V8 object, integer fields (fresh shape)	1.50	1.78×	0.94×	74.6 GB
V8 array of double arrays	1.88	2.22×	1.18×	93.3 GB
V8 object, double fields (reference)	3.50	4.14×	2.20×	174.1 GB
JVM class, boxed Double	3.81	4.52×	2.39×	189.6 GB
CPython __slots__ class	4.63	5.48×	2.90×	230.4 GB
CPython dataclass	5.38	6.37×	3.38×	267.7 GB
V8 Map	6.75	7.99×	4.23×	335.7 GB
CPython dict	7.38	8.74×	4.63×	367.1 GB
JVM HashMap<String, Double>	9.06	10.73×	5.68×	450.7 GB

*With $\alpha_{GC} = 0.25$. The earlier working range, β from 3.2 to 4.1, corresponds to W from 3.79×

Against an **off-heap** packed buffer (Float64Array, NumPy, a C array), which carries no collector headroom, $W = 1/1.126 = 0.89\times$. ATESO is 11% larger there. Against data that is already packed, ATESO's advantage is not density. It is what the 12.6% metadata buys: addressing, integrity and synchronization.

4. Topology, Partitioning and Fault Tolerance

4.1 The single-node form

The law applies per machine. Any host, whether Apple Silicon, AMD EPYC or AWS Graviton, holds W times more information in ATESO than in the reference representation, in the same state-bearing DRAM.

A 256 GB server with an 8 GB reserve holds 220.3 GB of information as resident granules. That is the equivalent of 1,027.8 GB of managed heap.

4.2 The reference cluster

The cluster has two hosts, two address spaces and one partitioned state space. Its capacity obeys the parent paper’s bound [2, §2.2]: each host’s shard must fit in that host’s memory minus its reserve.

$$|\mathcal{S}(i)| \cdot 72.0625 \text{ B} \leq M_i - M_i^{\text{reserved}}, \mathcal{S}(0) \cap \mathcal{S}(1) = \emptyset \quad \big|\mathcal{S}^{\{i\}}\big| \cdot 72.0625 \text{ B} \leq M_i - M_i^{\text{reserved}}, \quad \mathcal{S}^{\{0\}} \cap \mathcal{S}^{\{1\}} = \varnothing$$

	Node 1: M5 Max	Node 2: M1 Pro	Cluster
Role	Deliberative planner, state admission root	Deterministic reflex muscle	—
Nominal memory	36 GB	16 GB	52 GB
Declared reserve	6 GB	4 GB	10 GB
Arena	30 GB	12 GB	42 GB
Granules	416,305,290	166,522,116	582,827,406
Information held	26.644 GB	10.657 GB	37.301 GB
Managed-heap equivalent	124.3 GB	49.7 GB	174.1 GB

- **Reserve.** The reserve covers macOS, the planner and executor processes, .spine rings and log buffers. It is a declared configuration. Per the parent protocol [2, Appendix A.1], it is to be replaced by each host’s measured resident-set limit.
- **Units.** Figures of record are decimal, following the parent paper’s conservative convention. The hardware is binary: $52 \times 2^{30} \text{ B} = 55.83 \times 10^9 \text{ B}$. The 3.83 GB difference is unclaimed headroom on top of the declared reserve.
- **GPU residency.** If the arena is bound to the GPU as Metal buffers, it must also fit under the device’s recommended working-set size, which macOS sets below physical memory [A4].
- **Ceiling.** With no reserve at all, the ceiling would be $52 \times 4.1445 = 215.5 \text{ GB}$. No real host reaches it.

4.3 Credit-based flow control

- Each directed link carries a single-producer/single-consumer ring (.spine) of $65,536 \text{ slots} \times 64 \text{ bytes} = 4 \text{ MiB}$. Across hosts the ring is a protocol, not shared memory: the link carries explicit messages [2, §2.2].
- The consumer returns credits as it frees slots. The producer never spins and never overwrites an unconsumed slot. At zero credits it parks and propagates backpressure to the admission root. This is the credit discipline of ATM and InfiniBand fabrics [10].
- **Evidence.** Single-host backpressure and 32-bit counter wraparound were verified: 600,000 records across ring capacities of 2, 64 and 1,024 slots, every

field checked (R, parent artifact). Credits across the Thunderbolt link are specified but not yet measured (P).

4.4 Ownership and partitions

- Shards are disjoint and statically owned. DRAM holds no replicas, so the arithmetic in §5 counts every granule once.
- If the link drops, each node keeps serving its own shard. Cross-shard requests fail fast or queue under backpressure. Neither node ever writes the other's shard, so there is no split-brain to resolve and no quorum to hold.
- **The price.** There is no automatic failover. A failed node's shard stays unavailable until that node, or a replacement restored from its checkpoint and redo log, returns. Automatic failover would bring back fencing or consensus, because a silent peer cannot be told apart from a partitioned one [13].

4.5 Differential Merkle resynchronization

- Each shard keeps a binary Merkle tree [14] over 64 KiB leaves (1,024 granules) with 32-byte digests. The tree costs 0.098% of payload and is at most 19 levels deep on this cluster. Dirty leaves are rehashed at epoch boundaries, not on every write.
- A shard may have a remote copy, either a mirror or a shipped checkpoint image. When it does, reconnection reconciles the two. The nodes exchange roots and descend only divergent subtrees. At each divergent leaf they compare per-granule digests, computed on demand, and ship only the granules that differ.
- Cost follows divergence, not state size:

bytes on the wire $\approx 64 + 72\Delta + D(64\lceil \log_2 L \rceil + 8, 192)$ $\text{\texttt{bytes on the wire}}$ $\approx$;
 $64 + 72\Delta + D(64\lceil \log_2 L \rceil + 8, 192)$

Here Δ is the number of divergent granules, D the number of divergent leaves and L the number of leaves; message framing is excluded. Clustered divergence touches few leaves and is cheap. When most of a leaf has diverged, shipping the whole 64 KiB leaf costs less than the digest exchange, and the protocol does that instead. This is the anti-entropy pattern of Dynamo [12], applied at granule resolution.

4.6 Crash consistency through redo logging

- Every mutation appends a redo record before it is acknowledged. A full-image record holds the granule index (8 B), log sequence number (8 B), new image (64 B) and CRC32C (4 B): 84 bytes, padded to 96. A sparse byte-range delta may replace the full image.
- Checkpoints never read the live arena. They advance the on-disk image by replaying the log onto it, so concurrent writers cannot tear a checkpoint.
- Recovery loads the last checkpoint and replays every record after its sequence number. Replay is idempotent, and the checksum rejects a torn tail. This is ARIES-style redo-only recovery [7].
- On macOS, `fsync` does not flush the drive's write cache. Durability requires `fcntl(F_FULLFSYNC)` [A5].

- The log is circular. Its head may not overrun records newer than the last durable checkpoint. When it would, writers receive backpressure through the same credits as §4.3.

§4.4–4.6 are design specifications (P). They become results when the author’s implementation and traces join the reproduction package.

5. The Arithmetic Ledger

5.1 Micro-benchmark tier (N = 50,000)

- Payload: $50,000 \times 64 = 3,200,000$ B. Directory: 400,000 B (12.5%). Merkle tree: 49 leaves, 3,104 B (0.097%).
- Root history: at $H = 100$ it takes 3,200 B and $f = 0.1270$; at $H = 800$ it takes 25,600 B and $f = 0.1340$. A fixed history weighs heavily on a 3.2 MB state and vanishes at cluster scale (§5.2). The density law is stated at cluster scale.
- **Dirty-page locality (parent result).** A 5% mutation set scattered at random dirties an expected 752.25 of 782 pages of 4 KiB. The same mutations in 5 contiguous clusters dirty 44. That is a 94.15% reduction (F, analytical). The parent’s WebGPU benchmark run measured 94.2%: 744 pages down to 43 (R, parent artifact). The count of dirty granules is unchanged; locality removes page amplification.

5.2 Cluster tier (exact to the byte)

Bytes	Node 1	Node 2	Cluster
Arena	30,000,000,000	12,000,000,000	42,000,000,000
Granules N	416,305,290	166,522,116	582,827,406
Payload ($N \times 64$)	26,643,538,560	10,657,415,424	37,300,953,984
Directory ($N \times 8$)	3,330,442,320	1,332,176,928	4,662,619,248
Merkle tree	26,019,104	10,407,648	36,426,752
Leaves / depth	406,549 / 19	162,620 / 18	—
Unallocated	16	0	16
Managed-heap equivalent	124.337 GB	49.735 GB	174.071 GB

- The ledger closes: $37,300,953,984 + 4,662,619,248 + 36,426,752 + 16 = 42,000,000,000$ B.
- Metadata ratio on the cluster: $f = 0.1259766$. The 800-root history adds 25,600 B, about 0.00005% of 52 GB.
- Siege Mode (padded): 308,539,944 granules, 19.747 GB of information, 92.2 GB managed-heap equivalent.

5.3 Reserve sensitivity

Declared reserve	Arena	Sovereign equivalent	Siege equivalent
6 GB	46 GB	190.6 GB	101.0 GB
8 GB	44 GB	182.4 GB	96.6 GB
10 GB	42 GB	174.1 GB	92.2 GB
12 GB	40 GB	165.8 GB	87.8 GB
14 GB	38 GB	157.5 GB	83.4 GB

5.4 Units

- **Figures of record** are decimal: 1 GB = 10^9 B, the parent paper’s conservative convention [2, §2.2].
- **Hardware** is binary. The M5 Max reports `hw.memsize` = 38,654,705,664 B = 36×2^{30} B, and the cluster’s physical total is 55,834,574,848 B.
- **Rings and buffers** are binary: 4 MiB = 4,194,304 B.

6. Scope and Limits

- **The law is conditional on β .** It gives $4.14\times$ against the reference representation, $1.55\times$ against JVM classes with primitive fields and $10.7\times$ against per-record hash maps (§3.4).
- **Flat buffers are the boundary.** Against an off-heap packed buffer, $W = 0.89\times$. ATESO claims no density over data that is already packed. It claims a discipline that keeps data packed by construction, plus the addressing, integrity and synchronization its 12.6% buys.
- **Prior art.** Packed layouts are not new. Slab allocation [5], Apache Arrow [A7], FlatBuffers, Cap’n Proto, data-oriented design and off-heap stores all exploit $\beta \approx 1$, and the cost of bloat in managed heaps is well documented [4]. ATESO’s contribution is a runtime in which the packed layout is the only layout. It joins that layout to capability addressing, bounded flow control and verifiable synchronization.
- **Fragmentation.** Fixed 64-byte granules in uniform arenas have zero external fragmentation. A record smaller than 64 bytes wastes the remainder of its granule. A larger record spans granules and wastes up to 63 bytes in the last one. The law as stated assumes 64-byte records.
- **Mutation locality.** Epoch rehashing touches only dirty leaves. Spatially clustered mutation dirties few leaves. Uniformly scattered mutation at 5% dirties nearly every 64 KiB leaf, so each epoch approaches a full-state rehash.
- **Availability.** There is no automatic failover (§4.4).
- **GPU residency.** Metal’s recommended working-set size caps GPU-bound arenas below physical memory (§4.2).
- **Hardware class.** The cluster comparison prices capacity, not reliability (§1.2).
- **Prices move.** Every market figure in §1.1 carries its source and date.

7. Conclusion

The memory wall is a fabrication problem. For object-graph workloads it is also, and far more cheaply, a representation problem.

A plain JavaScript object spends 224 bytes to hold 64 bytes of numbers, and its collector then asks for a third more. ATESO spends 72. That ratio, measured rather than assumed, is the whole of the 4x multiplier.

On the same server, it delivers 4.14× more working set for the same money. On two owned Macs, it turns 52 GB of silicon into 174 GB of managed-heap equivalent, with no rent due.

The law is exact, and so are its limits. It rises with bloat, falls to 1.55× against disciplined JVM classes and inverts against a flat buffer. Within its domain of object graphs, document-shaped records and agentic state trees, resident execution is the law.

Memory has defected.

Appendix A: Measured β

Protocol. One million records per case. Each record holds eight float64 values (64 bytes of information). Live bytes were read after a full collection.

- V8: heapUsed plus external delta, Node.js 22.22.2, V8 12.4.254.21, pointer compression off.
- CPython 3.11.15: tracemalloc, NumPy 2.4.4.
- JVM: used-heap delta, OpenJDK 21.0.10, SerialGC, fixed 3 GB heap, compressed references.

The host was Linux x86-64, measured 22 September 2026. Sixty-four-bit object layouts in all three runtimes do not depend on the instruction set. Re-measurement on the arm64 reference hosts is scheduled (P).

Runtime	Representation	Bytes per record	β
V8	Object literal, 8 double fields	224.01	3.500
V8	Class instance, 8 double fields	224.01	3.500
V8	JSON.parse, 8-field document	224.01	3.500
V8	Object literal, 8 small integers, fresh shape	96.01	1.500
V8	Same, after the shape stored doubles	223.23	3.488
V8	Map, 8 string keys	432.01	6.750
V8	Array of 8-element double arrays	120.00	1.875
V8	Float64Array, packed	64.00	1.000
JVM	Class, 8 primitive double fields	84.00	1.313
JVM	Record, 8 double components	84.00	1.313

Runtime	Representation	Bytes per record	β
JVM	Class, 8 boxed Double fields	244.00	3.813
JVM	HashMap<String, Double>, 8 keys	580.00	9.063
JVM	double[], packed	64.00	1.000
CPython	Class with __slots__	296.45	4.632
CPython	Plain class	344.45	5.382
CPython	Dataclass	344.45	5.382
CPython	Tuple of 8 floats	304.45	4.757
CPython	Dict, 8 string keys	472.45	7.382
CPython	array('d'), packed	64.30	1.005
CPython	NumPy (N, 8) float64	65.35	1.021

Anatomy of the reference record (V8, 224 bytes):

Part	Bytes
Object header: map, properties and elements pointers	24
8 tagged slots	64
8 heap-number boxes $\times$ 16	128
Array slot	8
Total	224

Reproduction. The files are in RESEARCH/STAGE5-BETA-BENCHMARK/.

```
node --expose-gc bloat_v8.mjs
node --expose-gc bloat_v8_smi.mjs
python3 bloat_py.py
javac Bloat.java && java -XX:+UseSerialGC -Xms3g -Xmx3g Bloat
```

Appendix B: Claim Register

Claim	Value	Class	Where
Working-set law	$W \geq \beta / ((1 - \alpha)(\gamma + f))$	F	§3.1
Reference bloat factor	$\beta = 3.500$	R	App. A
Collector headroom	$\alpha = 0.25$, conservative	Stated assumption	§2.2, [6]
ATESO metadata	$f = 0.12598 \leq 0.126$	F	§2.4
Sovereign multiplier	4.14×	F	§3.2
	2.20×	F	§3.3

Claim	Value	Class	Where
Siege multiplier			
Multiplier vs off-heap packed buffer	0.89×	F	§3.4
Apple Silicon coherence granule	128 B	V	§3.3, [A2, A3]
Reference hosts	M5 Max 36 GB (32-core GPU bin); M1 Pro 16 GB	V + U	[2]
Declared reserve	6 GB + 4 GB	P	§4.2
Cluster managed-heap equivalent	174.1 GB	F, given the reserve	§5.2
Server with ATESO, managed-heap equivalent	1,027.8 GB	F	§4.1
Server price	\$15,200	U	§1.2
Mac prices	\$3,599 + \$1,999	V	[A1, A6]
Q1 2026 conventional DRAM contract rise, realized	+93–98% QoQ	V	[T7]
Q1 2026 server DRAM contract rise, forecast	≈ +90% QoQ	V	[T1]
64 GB RDIMM Q1 2026 contract	\$820–950	V	[T2]
64 GB RDIMM Q4 2026 projection	\$1,586	V	[T3]
32 GB DDR5, Sept → Nov 2025	\$149 → \$239	V	[T4]
Q3 2026 server DRAM contract rise	+13–18%	V	[T5]
	≈ 18% → ≈ 53%	V	[T6]

Claim	Value	Class	Where
Memory share of a 512 GB reference build			
.spine backpressure and counter wrap	600,000 records	R	§4.3
Cross-host credit flow control	—	P	§4.3
Static ownership, no failover	—	P	§4.4
Differential Merkle resync	—	P	§4.5
Redo-log crash consistency	—	P	§4.6
Dirty-page reduction under clustering	94.15% / 94.2%	F / R	§5.1

Appendix C: Symbols

Symbol	Meaning
S	Information held, bytes (64 per granule)
β	Live managed bytes per byte of information
α_{GC}	Free fraction a collector keeps in its heap
γ	Slots per granule (1 packed, 2 padded)
f	ATESO metadata bytes per payload byte
W	Working-set multiplier, $M_{\text{heap}} / M_{\text{ATESO}}$
N	Granule count
L, D, Δ	Merkle leaves, divergent leaves, divergent granules
$M_i, M_i^{\text{reserved}}$	Host memory and its operating reserve

References

Parent research

1. B. DeCrow. *ATESO & The Magma Runtime: The Thermodynamic Obsolescence of Document-Centric Execution for Continuous Spatial Compute*. ManyMoats Research, 20 Sept 2026. USPTO App. 64/159,586.
2. B. DeCrow. *Capability-Addressed Binary State and Zero-Heap Resident Memory Vectors for Hard Real-Time Physics and Closed-Loop Agentic Coordination*. Independent preprint, v2, 21 Sept 2026.

Systems literature

1. G. M. Amdahl. Validity of the single processor approach to achieving large scale computing capabilities. *AFIPS Spring Joint Computer Conference*, 1967.
2. N. Mitchell, G. Sevitsky. The causes of bloat, the limits of health. *OOPSLA*, 2007.
3. J. Bonwick. The slab allocator: an object-caching kernel memory allocator. *USENIX Summer Technical Conference*, 1994.
4. M. Hertz, E. D. Berger. Quantifying the performance of garbage collection vs. explicit memory management. *OOPSLA*, 2005.
5. C. Mohan, D. Haderle, B. Lindsay, H. Pirahesh, P. Schwarz. ARIES: a transaction recovery method supporting fine-granularity locking and partial rollbacks using write-ahead logging. *ACM TODS* 17(1), 1992.
6. I. Sheludko, S. Aboy Solanes. Pointer compression in V8. *v8.dev blog*, 30 March 2020.
7. W. J. Bolosky, M. L. Scott. False sharing and its effect on shared memory performance. *USENIX SEDMS IV*, 1993.
8. H. T. Kung, R. Morris. Credit-based flow control for ATM networks. *IEEE Network* 9(2), 1995.
9. M. Macklin, M. Müller, N. Chentanez. XPBD: position-based simulation of compliant constrained dynamics. *Motion in Games (MIG)*, 2016.
10. G. DeCandia et al. Dynamo: Amazon's highly available key-value store. *SOSP*, 2007.
11. S. Gilbert, N. Lynch. Brewer's conjecture and the feasibility of consistent, available, partition-tolerant web services. *SIGACT News* 33(2), 2002.
12. R. C. Merkle. A digital signature based on a conventional encryption function. *CRYPTO '87*, 1987.

Market data

- [T1] TrendForce. Memory price outlook for 1Q26 sharply upgraded; QoQ increases of all product categories to hit record highs. 2 Feb 2026. <https://www.trendforce.com/presscenter/news/20260202-12911.html>
- [T2] MemoryMarket (CFM). CFM reports latest DDR5 RDIMM prices, exceeding 100% increase compared to last quarter. 10 Feb 2026. <https://www.memorymarket.com/a/14593>
- [T3] Servnet UK, citing Citi (May 2026). DDR5 server memory price 2026: the 90% quarter tracked. <https://www.servnetuk.com/research/ddr5-server-memory-price-tracker-2026>
- [T4] Network World, citing Reuters. Samsung's 60% memory price hike signals higher data center costs for enterprises. 17 Nov 2025. <https://www.networkworld.com/article/4091014/samsungs-60-memory-price-hike-signals-higher-data-center-costs-for-enterprises.html>

- [T5] TrendForce. Long-term agreements cap price increases; server DRAM contract prices expected to rise 13–18% QoQ in 3Q26. 9 Jul 2026. <https://www.trendforce.com/presscenter/news/20260709-13140.html>
- [T6] Servnet UK. Server prices 2026: how memory repriced the rack. <https://www.servnetuk.com/research/enterprise-server-price-index-2026>
- [T7] TrendForce. Rapid contract price surge drives 1Q26 DRAM industry up 81% QoQ. 1 Jun 2026. <https://www.trendforce.com/presscenter/news/20260601-13070.html>

Vendor and platform sources

- [A1] Macworld. MacBook Pro M5 Pro & Max 2026 complete guide: price, specs, release date (14-inch M5 Max, 36 GB / 2 TB, from \$3,599). <https://www.macworld.com/article/2942089/macbook-pro-m5-pro-max-release-specs-price.html>
- [A2] ERGO-Code/HiGHS issue #3221: `sysctl hw.cachelinesize` reports 128 on Apple silicon. <https://github.com/ERGO-Code/HiGHS/issues/3221>
- [A3] 7-cpu.com. Apple M1: Firestorm L1D 64 B/line; L2 128 B/line. https://www.7-cpu.com/cpu/Apple_M1.html
- [A4] Apple Developer Documentation. `MTLDevice.recommendedMaxWorkingSetSize`.
- [A5] Apple. `fsync(2)` manual page: `F_FULLFSYNC`.
- [A6] Apple Newsroom. Apple unveils game-changing MacBook Pro (14-inch from \$1,999). 18 Oct 2021.
- [A7] Apache Arrow. Columnar format specification.