The Cache Wars: An Empirical Study of Prompt-Cache Efficiency in Six LLM Agent Harnesses

Annie

AGNT Labs

Technical Report · v2.0 · July 7, 2026

Abstract

LLM agent harnesses re-transmit the full conversation context on every model invocation, making input cost quadratic in session length. Provider-side prompt caching mitigates this by discounting previously-processed prefix tokens by 90%, but only if the harness (i) emits cache-control markers, (ii) selects a time-to-live (TTL) that survives realistic human pauses, and (iii) keeps the request prefix byte-stable across turns. We present a controlled measurement of prompt-cache efficiency across six production agent harnesses — AGNT v0.6.4, oh-my-pi (OMP) 16.3.11, Claude Code CLI 2.1.126, Codex CLI 0.130.0, OpenClaw 2026.5.12, and Hermes Agent 0.18.0 — using byte-identical workloads, unmodified release binaries, and per-request billing counters obtained from provider telemetry via transparent logging proxies with fail-loud validity guards. The central instrument is a *pause test*: four rapid turns followed by a 390-second idle period that exceeds Anthropic's 5-minute cache TTL but not its 1-hour TTL. We find a 4.3–9.8× divergence in post-pause billing: harnesses emitting 1-hour markers (AGNT, Claude Code) retained 86.0–86.5% of the request in cache, while harnesses emitting 5-minute markers (OpenClaw, Hermes) retained 0% and re-purchased their entire context at a 1.25× write premium. We additionally identify two previously unreported cost pathologies: (1) Claude Code injects 6,817 tokens of self-generated scaffolding per turn, inflating steady-state premium spend 2.3× over the user payload; and (2) Hermes' cache-marker placement causes a monotone leak in which cached reads shrink while premium writes grow within a single burst. We further show that zero self-injection overhead is not unique to AGNT: OMP bills premium tokens equal to the user payload exactly, yet ships a 5-minute cache default under API-key authentication (1-hour only via OAuth or the `PI_CACHE_RETENTION=long` flag), so it fails the same pause test unless reconfigured — demonstrating that payload-clean billing and a survivable default TTL are independent properties. Extrapolated to a one-hour, 20-message session with four natural breaks, measured behavior yields totals of $1.07 (AGNT), $1.51 (OMP, shipping default), $1.98 (OpenClaw), $2.09 (Hermes), and $2.32 (Claude Code) against a $4.02 uncached ceiling. Composed over a working month (22 days, 8 h/day), the same measured parameters project $120/month (AGNT) versus $325–$410/month for the other Anthropic-path harnesses and $708 uncached — a per-seat "cache tax" of $205–$290/month, or $2,500–$3,500/year, attributable entirely to client-side marker engineering. All artifacts, scripts, and raw counters are provided for independent replication.

Keywords: prompt caching, LLM agents, agent harness, cost measurement, cache TTL, context management, reproducibility

1 Introduction

Autonomous and semi-autonomous LLM agents operate in a loop: the harness assembles a request containing a system prompt, tool schemas, the full conversation history, and the newest user message; the model responds; the cycle repeats. Because the entire context is re-transmitted on every call, cumulative input tokens grow quadratically with turn count. For multi-hour, tool-heavy sessions this term dominates total cost.

Anthropic and OpenAI both offer prompt caching: previously processed prefix tokens are re-served at a fraction of list price (10% on Anthropic [1]; free discounting on OpenAI [2]). Effectiveness, however, is entirely delegated to the client harness. Three client-side decisions determine realized savings: (a) whether cache-control markers are emitted at all; (b) the TTL selected per marker (Anthropic: 5 minutes at a 1.25× one-time write premium, or 1 hour at 2× [1]); and (c) whether the serialized prefix remains byte-identical across turns, since provider caches are exact-prefix matches — any earlier-byte mutation invalidates everything after it.

Vendor documentation describes intended behavior; it does not establish what shipped binaries do under realistic use. This paper measures that directly. Our contributions:

- A reproducible **pause-test protocol** that cleanly discriminates 5-minute from 1-hour cache retention using a 390 s idle gap.
- Per-turn billing telemetry for six production harnesses on byte-identical workloads, with instrumentation guards that abort on any validity violation.
- Identification of two cost pathologies invisible to documentation review: per-turn scaffolding injection (Claude Code, +6,817 tokens/turn) and intra-burst cache leakage (Hermes).
- A parametric cost model, fit to the measurements, projecting one-hour and multi-hour session costs, and its composition into monthly and annual per-seat spend under three usage profiles.
- A complete artifact set enabling third-party replication.

2 Background: Prompt-Cache Semantics and Pricing

Anthropic. A request may carry up to four `cache_control` breakpoints. Marking a content block caches the serialized request prefix up to and including that block. Each marker carries a TTL: `ephemeral` 5 m (default) or 1 h (requires the `extended-cache-ttl-2025-04-11` beta header). Billing for Claude Sonnet 4.5 (list, per 10^6 tokens): base input $3.00; cache read $0.30 (0.1×); 5 m cache write $3.75 (1.25×); 1 h cache write $6.00 (2.0×) [1]. Cache hits refresh the TTL. Caching is exact-prefix: a single differing byte at position N invalidates all cached content at positions $\geq N$.

OpenAI. Caching is automatic and unpriced: requests sharing a prefix ≥1,024 tokens may report `cached_tokens` at a discount. There are no client-visible markers, no TTL selection, and no retention guarantee; eviction is load-dependent, typically minutes-scale [2]. Codex CLI relies exclusively on this mechanism [3].

We use *premium tokens* to denote tokens billed at ≥1× list (uncached input plus cache writes) and *cached tokens* for those billed at 0.1×. For a harness with zero overhead, steady-state premium per turn equals the size of the newest message.

3 Systems Under Test

Table 1. Harnesses, versions, installation channel, and measured provider path.

Harness	Version	Install	Provider path measured	Marker TTL (measured)
AGNT	v0.6.4	release build	Anthropic direct (API key)	1 h (all four breakpoints)
oh-my-pi (OMP)	16.3.11	bun i -g @oh-my-pi/pi-coding-agent	Anthropic direct (API key)	5 m default; 1 h via env/OAuth
Claude Code CLI	2.1.126	npm (official)	Anthropic (native)	1 h (all writes)
Codex CLI	0.130.0	npm (official)	OpenAI (native)	n/a — automatic
OpenClaw	2026.5.12	npm i -g openclaw	Anthropic direct (API key)	5 m (all writes)
Hermes Agent	0.18.0	pip install hermes-agent (Py 3.12)	Anthropic direct (API key)	5 m (self-reported & observed)

All binaries are unmodified releases. Anthropic-path harnesses used model `claude-sonnet-4-5-20250929` (Claude Code additionally routes sub-tasks to Haiku per its defaults). OpenClaw and Hermes were driven through their public entry points (`openclaw agent --local`; Hermes' documented `AIAgent` API with `run_conversation(…, conversation_history=…)`). Each was configured on the provider path where its caching demonstrably engages — Hermes' runtime log confirms *"Prompt caching: ENABLED (native Anthropic, 5m TTL)"* — so results reflect each system's best case, not a degraded default. OMP was driven through its own `omp -p --mode json --continue` print mode; its per-turn `agent_end` usage block reports a native per-TTL split (`cttl:{ephemeral5m, ephemeral1h}`), and it was measured on both its API-key default (5 m) and its `PI_CACHE_RETENTION=long` path (1 h).

4 Experimental Design

4.1 Workload

Each turn *t* sends the message `"TURN t: Reply with exactly OKt. Ignore: F(t)"`, where the filler *F(t)* is a deterministic function producing ≈5,186 tokens (Listing 1). Payloads are byte-identical across harnesses. Instructing single-token replies (`OKt`) minimizes output-side confounds.

```
Listing 1 — deterministic per-turn filler (JavaScript; Python port identical)
function filler(t) {
  let s = '';
  for (let i = 0; i < 260; i++)
    s += `row ${t}-${i} a=${(i*7919)%104729} b=${(i*104729)%7919} c=${i%13}; `;
  return s;
}
```

4.2 Protocol: the pause test

Four turns are issued back-to-back (inter-turn latency ≪ 5 m), followed by an idle period of 390 s, followed by turn 5. The idle length is chosen to strictly exceed the 5-minute TTL while remaining far below the 1-hour TTL, so post-pause telemetry classifies each harness's effective retention with no ambiguity: a 5 m cache must report `cache_read_input_tokens = 0` on turn 5; a 1 h cache must report a full-prefix read.

4.3 Instrumentation and validity guards

Ground truth is the provider's own per-request accounting: Anthropic's `usage` block (`input_tokens`, `cache_read_input_tokens`, `cache_creation_input_tokens`, and the per-TTL split `cache_creation.ephemeral_{5m,1h}_input_tokens`); Codex's session-log `last_token_usage` events; Claude Code's `--output-format json` result blocks. For OpenClaw and Hermes we additionally interposed a transparent localhost logging proxy that records, per call, the full set of `cache_control` markers in the outgoing request and the usage block of the response, without modifying either.

Driver scripts enforce three fail-loud invariants; violation aborts the run and discards its data: (G1) *routing* — every turn must append exactly ≥1 new proxy log line (defeats silent proxy bypass); (G2) *growth* — the request's message count must increase monotonically across turns (defeats non-accumulating history; Hermes verified at 1→3→5→7→9 messages); (G3) *freshness* — for log-scraped harnesses, the consumed telemetry event must carry a timestamp newer than the previous turn's (defeats stale-read errors).

4.4 Cross-harness comparability

Harnesses ship system prompts and tool schemas of different sizes (12.6 k–32.8 k tokens at turn 1), so absolute token counts are not comparable across systems. We therefore compare only: (i) the post-pause **cache-hit ratio** (dimensionless); (ii) **steady-state premium tokens per turn** against the known 5,186-token payload, which decomposes each bill into *user payload* + *harness overhead*; and (iii) dollar figures from a **controlled replay** in which each harness's marker strategy is applied to fully identical content (identical synthetic system prompt, identical four tool schemas, identical fillers) against the live Anthropic API, each arm carrying a unique isolation key to prevent cross-arm cache pollution.

As an internal consistency check, per-turn context growth must equal the payload plus any harness self-injection. Measured deltas (turns 2–4, tokens/turn): AGNT +5,186; OMP +5,186; Hermes +5,186; OpenClaw +5,189; Codex +7,485; Claude Code +12,003. The constant ≈5,186 component appearing in all six confirms workload identity to within tokenizer noise; the excess (+2,299 Codex, +6,817 Claude Code) is measured harness overhead (§5.4).

5 Results

5.1 Per-turn billing telemetry

Table 2. Per-turn provider counters (tokens). C = cached (billed 0.1×); P = premium (billed ≥1×). Turn 5 follows the 390 s pause

Turn	AGNT v0.6.4 C	P	Claude Code C	P	Codex C	P	OpenClaw C	P	Hermes C	
1	0	17,809	18,537	19,469	2,432	18,677	0	29,982	0	
2	17,809	5,186	38,006	12,003	20,864	7,730	29,982	5,189	22,837	
3	22,995	5,186	50,009	12,003	28,544	7,535	35,171	5,189	17,339	
4	28,181	5,186	62,012	12,003	35,712	7,852	40,360	5,189	17,339	
5 (post-pause)	**33,367**	**5,191**	**74,015**	**12,009**	**43,392**	**7,659**	**0**	**50,743**	**0**	**4**

AGNT's per-TTL split reports `ephemeral_5m_input_tokens = 0` on every turn (all writes 1 h). Claude Code's CLI-reported per-turn costs were $0.1405, $0.0942, $0.1002, $0.1062, $0.1123. Hermes' proxied requests carried four 5 m markers per call. OMP's counters (both TTL paths) are reported separately in §5.6 (Table 3a), since its default and opt-in configurations must be distinguished.

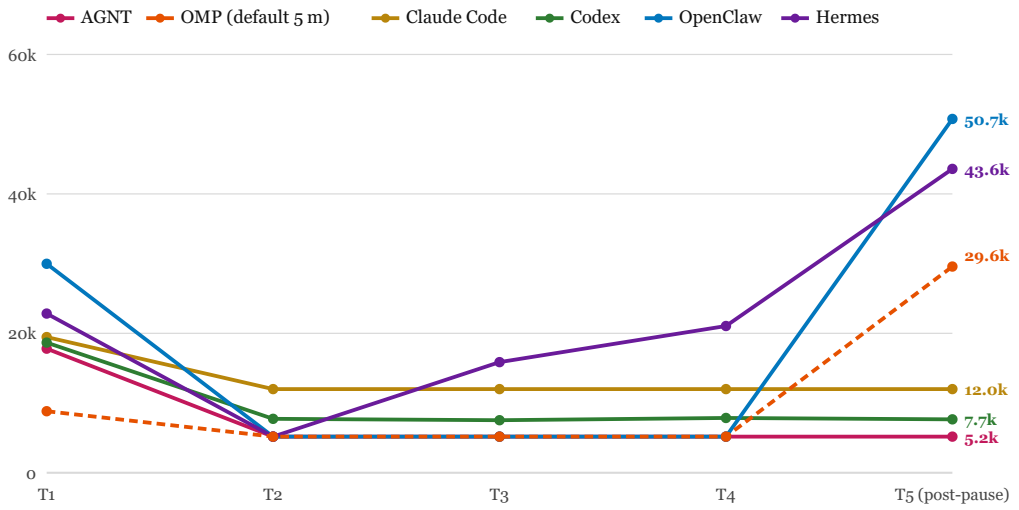

Figure 1. Premium input tokens billed per turn (rate ≥1× list). Turn 5 follows the 390 s idle: the 5-minute-TTL harnesses (OpenClaw, Hermes, and OMP on its API-key default) re-purchase their full accumulated context at the 1.25× write premium, while 1-hour harnesses resume at payload cost. OMP's premium is the payload alone through turn 4 (zero overhead), then its 5 m default expires at the pause. Hermes additionally climbs during the burst (§5.5).

5.2 Pause survival

Table 3. Turn-5 cache-hit ratio h = $C_5/(C_5+P_5)$.

Harness	TTL	C_5	P_5	h	Classification
AGNT v0.6.4	1 h	33,367	5,191	**86.5%**	survived; premium = payload only
OMP 16.3.11 (=long)	1 h	24,399	5,191	82.5%	survived (opt-in); premium = payload only
OMP 16.3.11 (default)	5 m	0	29,570	0%	expired; full-context re-write (API-key default)
Claude Code	1 h	74,015	12,009	86.0%	survived; premium = payload + overhead
Codex CLI	auto	43,392	7,659	85.0%	survived this trial; retention unguaranteed [2]
OpenClaw	5 m	0	50,743	0%	expired; full-context re-write at 1.25×
Hermes	5 m	0	43,583	0%	expired; full-context re-write at 1.25×

The result partitions exactly along declared TTL, consistent with Anthropic's contractual expiry semantics: both 5 m harnesses lost 100% of cache across the 390 s gap, re-billing 43.6–50.7 k tokens at the write premium, while both 1 h harnesses read their entire prior context at 0.1×. Codex's survival is a single observation of an explicitly unguaranteed mechanism and is not generalizable.

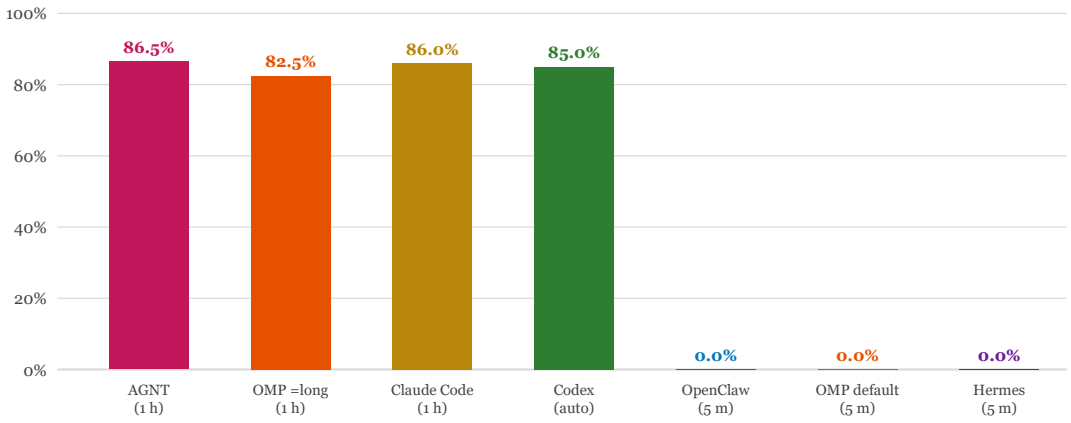

Figure 2. Turn-5 cache-hit ratio h = $C_5/(C_5+P_5)$ after the 390 s pause. The outcome partitions exactly along the effective TTL: 82.5–86.5% for 1-hour markers (including OMP with `PI_CACHE_RETENTION=long`), 0% for 5-minute markers (OpenClaw, Hermes, and OMP on its API-key default), with no intermediate value.

5.3 Controlled-replay cost (identical content)

Replaying marker strategies over identical content (§4.4) for the 5-turn protocol yields, at Sonnet 4.5 list prices: all-1 h strategy (AGNT, Claude Code layouts) **$0.262**; 5 m strategies (OpenClaw, Hermes layouts) **$0.290** (+11%, driven by the post-pause re-write); no-cache control **$0.423** (+61%). Verification: all-1 h writes 17,809 + 3×5,186 + 5,191 = 38,557 tk × $6/M = $0.231; reads 102,352 tk × $0.30/M = $0.031; total $0.262. ✓

5.4 Pathology I — per-turn scaffolding injection (Claude Code)

Claude Code's context grows 12,003 tokens/turn against a 5,186-token payload: 6,817 tokens/turn (2.31× payload) of self-injected content (system reminders and related scaffolding) is appended inside the billed conversation and re-billed at the 1 h write premium every turn. Its large cached column in Table 2 partially reflects caching of its own injected bloat; both its write premium and its 0.1× read base are inflated by it. Perfect marker discipline coexists here with the worst steady-state premium spend of the three surviving harnesses.

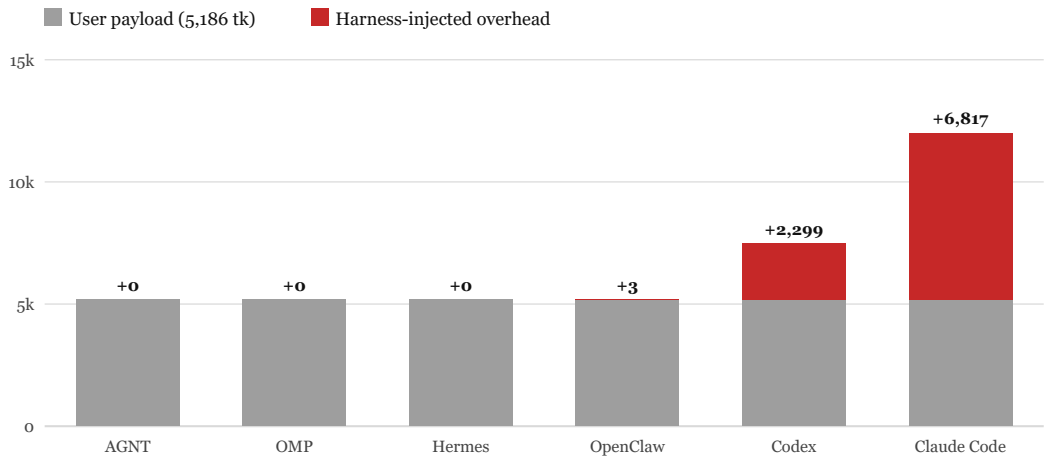

Figure 3. Per-turn conversation growth decomposed into the byte-identical 5,186-token user payload (gray) and harness-injected overhead (red). AGNT and OMP inject nothing — premium spend is the payload alone; Codex adds 2,299 tokens/turn and Claude Code 6,817 (2.31× the payload), each re-billed at the write premium every turn.

5.5 Pathology II — intra-burst cache leak (Hermes)

Within the rapid burst — before any TTL expiry — Hermes' cached reads *decrease* (22,837 → 17,339 from turn 2 to 3, then plateau) while premium writes *grow* (5,186 → 15,869 → 21,054). The read plateau at ≈17.3 k corresponds to its stable system/tool prefix; its rolling message-marker placement fails to extend the cached prefix over accumulated history, so a growing suffix of the conversation is re-written at 1.25× on every call. Hermes leaks cost even while its cache is nominally alive, then loses the remainder at the pause.

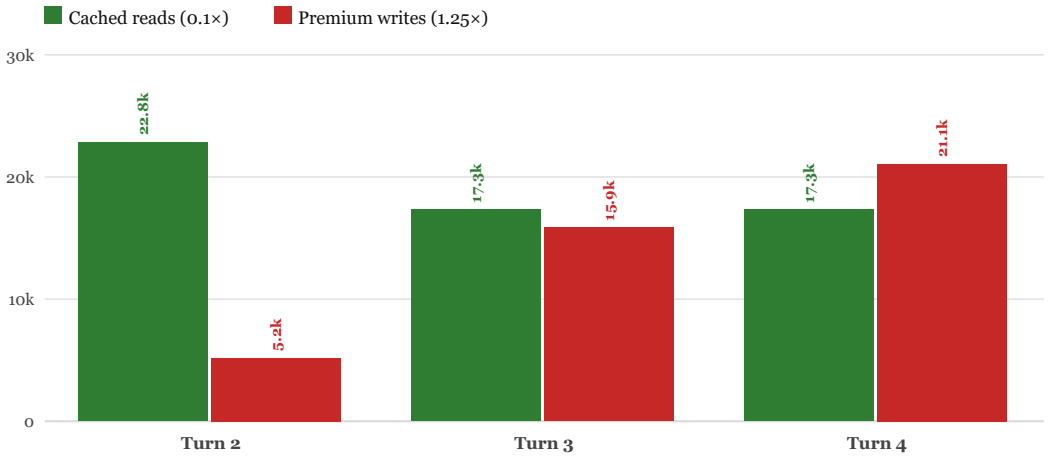

Figure 4. The Hermes intra-burst leak (Table 2 rows, turns 2–4, before any TTL expiry): cached reads collapse to the ≈17.3k static prefix while premium writes grow with the conversation, because rolling marker placement never extends the cached prefix over history.

5.6 OMP — clean spend, but a default-TTL trap

oh-my-pi is the only harness besides AGNT that bills premium tokens equal to the user payload with zero self-injection (5,186 tokens/turn, measured; Table 2-consistent delta +5,186). Its cache logic (`packages/ai/src/providers/anthropic.ts:424`) reads `retention = cacheRetention ?? (isOAuthToken ? "long" : resolveCacheRetention())`, and `resolveCacheRetention` returns `"short"` unless `PI_CACHE_RETENTION=long`. Thus under API-key authentication OMP emits 5-minute markers by default and

fails the pause test identically to OpenClaw and Hermes (h = 0%, full 29,570-token re-write; verified per-TTL as `ephemeral_5m`). With the opt-in flag (or an OAuth token) it emits 1-hour markers and survives (C_5 = 24,399, h = 82.5%, verified `ephemeral_1h`). OMP demonstrates that payload-clean billing and the 5-minute default trap are *independent* axes: a harness can perfect the former and still forfeit the pause on the latter.

Table 3a. OMP per-turn counters, both TTL paths (tokens). C = cached; P = premium. Premium is pinned at the payload every turn; only the default-vs-flag TTL differs.

Turn	Default (5 m) C	P	=long (1 h) C	P
1	0	8,821	0	8,841
2	8,821	5,186	8,841	5,186
3	14,007	5,186	14,027	5,186
4	19,193	5,186	19,213	5,186
5 (post-pause)	**0**	**29,570**	**24,399**	**5,191**

6 Cost Model and One-Hour Extrapolation

Let the session comprise n messages of payload m tokens, partitioned into bursts by breaks longer than the TTL. With prefix c_1 = first-turn write and per-turn overhead o, and prices r = \$0.30/M (read), w_5 = \$3.75/M, w_1 = \$6/M, u = \$3/M:

$$Cost_{1h} = w_1 \cdot [c_1 + (n-1)(m+o)] + r \cdot \Sigma_{t=2..n}\, prefix(t) \qquad Cost_{5m} = w_5 \cdot [\Sigma_{bursts}\, rewrite_b + in\text{-}burst\ writes] + r \cdot (in\text{-}burst\ reads)$$

Instantiated with measured per-harness parameters for **n = 20, m = 5,186, four breaks** (after messages 4, 8, 12, 16; breaks > 5 m, session ≤ 1 h):

Table 4. One-hour projection (measured parameters; Sonnet 4.5 list prices). M = uncached multiplier vs. \$4.02 control.

Harness	Premium writes (tk)	@$	Cached reads (tk)	@$	Total	M
AGNT v0.6.4 (1 h, o=0)	116,343	\$0.70	1,225,177	\$0.37	**\$1.07**	**0.27×**
OMP (default 5 m, o=0)	329,335	\$1.23	910,215	\$0.27	\$1.51	0.38×
OpenClaw (5 m; rewrites 357,367 + 77,835)	435,202	\$1.63	≈1,167k	\$0.35	\$1.98	0.49×
Hermes (5 m + leak ≈14k/turn)	≈531k	\$2.00	≈300k	\$0.09	\$2.09	0.52×
Claude Code (1 h, o=6,817)	247,526	\$1.49	2,793,164	\$0.84	\$2.32	0.58×
No caching (control)	1,341,520 @ u	\$4.02	—	—	\$4.02	1.00×

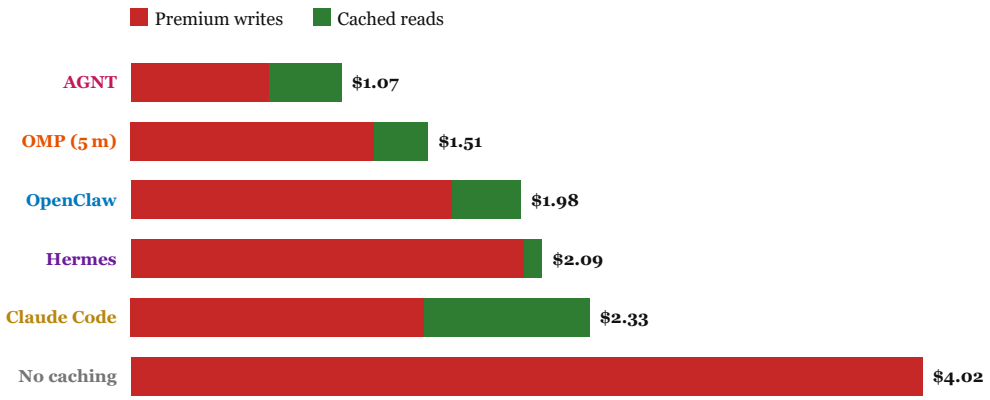

Figure 5. One-hour session projection (20 messages, 4 natural breaks; Table 4), decomposed into premium writes and cached reads. OpenClaw/Hermes/OMP-default spend on re-writes at every break; Claude Code pays both rates on its scaffolding; AGNT's premium spend is the payload alone. OMP incurs the break re-writes but zero overhead, so it sits between AGNT and OpenClaw.

Extending the same model to longer sessions (breaks every ≈15 min) yields the multipliers of Table 5. AGNT's is uniquely monotone decreasing — the 2× build premium amortizes into 0.1× reads. OpenClaw's and Claude Code's flatten: every break re-charges a full, ever-larger re-write (Claude Code additionally offset upward by o). Hermes' is non-monotone — the leak grows with context and eventually dominates. The model is conservative for 5 m harnesses: it assumes zero intra-burst expiry; real usage pauses more often than every 15 minutes, widening all gaps in AGNT's favor. OMP is reported on its shipping 5-minute default ($1.51, 0.38×); with the opt-in 1-hour flag its smaller system prompt (8.8k vs AGNT's 17.8k first-turn write) yields $0.96 (0.24×), marginally below AGNT — but that is not out-of-box behavior, so the default is used for the headline, consistent with the treatment of every other harness. Codex is excluded from dollar projections (OpenAI-only; different price list); in token terms it bills ≈7.7 k premium/turn versus AGNT's 5.2 k (+48% overhead).

Table 5. Cost multiplier vs. the uncached baseline by session length (lower is better).

Harness	15 m	30 m	1 h	2 h	4 h	Shape
AGNT v0.6.4	**0.62**	**0.40**	**0.27**	**0.21**	**0.17**	monotone decreasing
OMP (default 5 m)	0.58	0.44	0.38	0.36	0.35	flattens (clean, but 5 m)
OpenClaw	0.69	0.55	0.49	0.47	0.46	flattens
Hermes	0.72	0.60	0.52	0.55	0.58	non-monotone (leak)
Claude Code	0.75	0.64	0.58	0.54	0.52	flattens (offset by o)

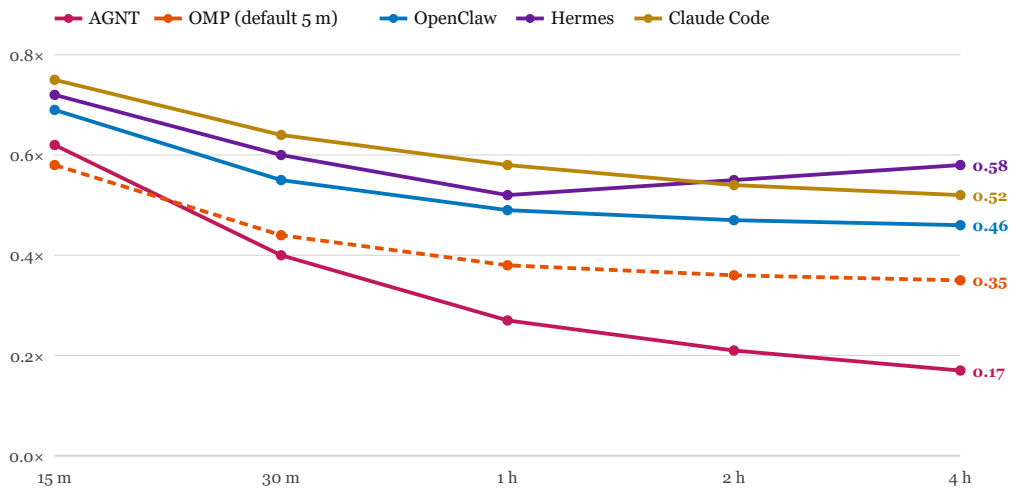

Figure 6. Cost multiplier vs. the uncached baseline by session length (Table 5; lower is better). AGNT's curve is uniquely monotone decreasing; OpenClaw, Claude Code, and OMP (5 m default) flatten; Hermes turns upward as the leak grows with context. OMP flattens lowest of the 5-minute-default harnesses because it carries no per-turn overhead — but it never reaches AGNT's trajectory without the 1-hour flag.

7 Monthly and Annual Extrapolation

The session-level model of §6 composes linearly across sessions: a month of usage is a sequence of independent sessions, each starting cold (first-turn cache build) and evolving per the measured per-harness parameters. Let R_u = \$4.02/h denote the uncached burn rate of the reference workload (20 messages/h at 5,186 tokens each) and $M_X(L)$ the session-length-dependent cost multiplier of harness X from §6 (Table 5). Monthly cost over D workdays with a fixed daily session schedule S is

$$Cost_X^{month} = D \cdot \Sigma_{L \in S} M_X(L) \cdot L \cdot R_u$$

We instantiate three usage profiles at D = 22 workdays (Table 6) and evaluate with the measured multipliers (Table 7).

Table 6. Usage profiles for monthly extrapolation.

Profile	Daily usage	Session shape	Multiplier basis
Light	2 h/day (44 h/mo)	2 × 1 h sessions	M(1 h)
Moderate	4 h/day (88 h/mo)	2 × 2 h sessions	M(2 h)
Heavy	8 h/day (176 h/mo)	2 × 4 h sessions	M(4 h)

Table 7. Projected monthly cost per seat (USD; 22 workdays; Sonnet 4.5 list prices).

Harness	Light	Moderate	Heavy
AGNT v0.6.4	**$47**	**$74**	**$120**
OMP (default 5 m)	$67	$127	$247
OpenClaw	$87	$166	$325
Hermes	$92	$195	$410
Claude Code	$102	$191	$368
No caching (control)	$177	$354	$708

Two structural effects emerge at monthly scale. First, **AGNT's advantage widens with load**: because its multiplier is monotone-decreasing in session length (0.27 at 1 h → 0.17 at 4 h) while the others flatten, its relative advantage grows from ≈1.9× (light) to 2.7–3.4× (heavy). Second, **the harness ranking inverts under load**: Claude Code is the most expensive surviving harness at light usage (scaffolding tax dominates short sessions), but Hermes overtakes it at the heavy profile because the intra-burst leak (§5.5) grows with context length — Hermes is the only harness whose per-token cost *rises* as sessions lengthen.

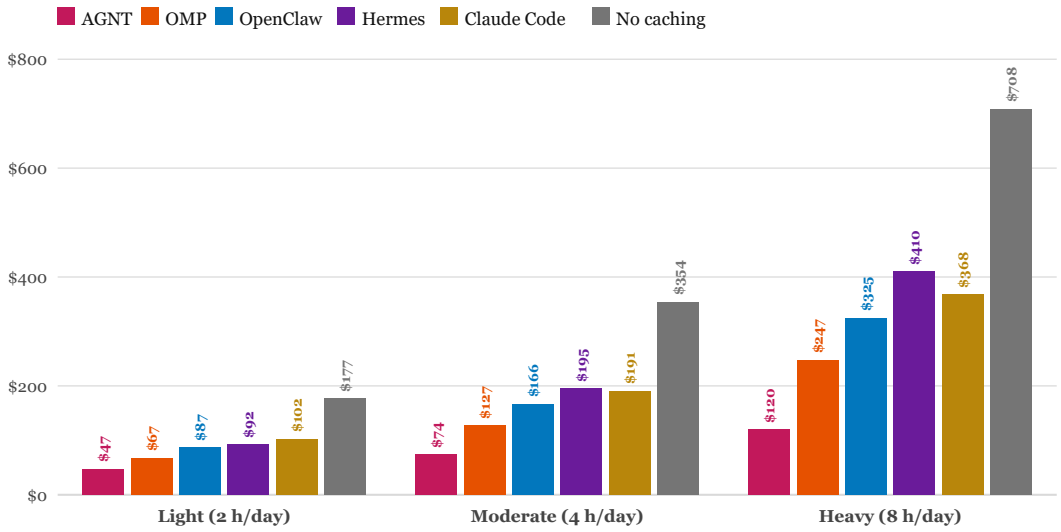

Figure 7. Projected monthly cost per seat under the three usage profiles of Table 6 (22 workdays; Sonnet 4.5 list prices). AGNT's advantage widens from ≈1.9× (light) to 2.7–3.4× (heavy); OMP (5 m default) is the cheapest of the non-AGNT harnesses at every profile thanks to zero overhead, but its 5-minute default keeps it well above AGNT; Hermes overtakes Claude Code as the most expensive harness at the heavy profile.

7.1 The cache tax

Define the *cache tax* of harness X as $\text{Cost}_X - \text{Cost}_{\text{AGNT}}$ for identical work on the identical model. At the heavy profile (Table 8), the tax is $205–$290 per seat per month; a heavy OpenClaw or Hermes seat spends AGNT's entire monthly budget by approximately day 8.

Table 8. Cache tax vs. AGNT v0.6.4, heavy profile (8 h/day, 22 workdays).

Harness	$/month	$/year	Dominant mechanism
OMP (default 5 m)	127	1,524	full-context re-write after every >5 m break (no overhead; 1 h available via flag)
OpenClaw	205	2,462	full-context re-write at 1.25× after every >5 m break
Claude Code	248	2,971	6,817 tokens/turn self-injected scaffolding at 1 h premium
Hermes	290	3,481	pause re-writes *plus* intra-burst prefix leak
No caching	587	7,046	all tokens at list price

7.2 Team scale

Annualized over a five-seat team at the heavy profile: AGNT $7,220; OpenClaw $19,528; Claude Code $22,075; Hermes $24,622; uncached $42,451. The Hermes-vs-AGNT delta alone is ≈$17,400/year for byte-identical work — the cost of one configuration bit and marker placement, not a model upgrade.

Three properties of this extrapolation bias it *against* the headline gap rather than for it: (i) the session model assumes breaks only every ≈15 minutes, whereas real interactive work pauses more often, adding re-write events that penalize only the 5 m-TTL harnesses; (ii) the Hermes figure holds its leak rate at the 4-hour-measured value, though the leak grows with context; and (iii) real payloads (tool results, file contents) are larger than the synthetic filler, scaling every re-write linearly. The figures for OpenClaw and Hermes are therefore lower bounds under the stated price list.

7.3 The agentic amplifier: tool latency as involuntary pause

The pause test models the idle gap as a *human* pause, but nothing in the mechanism requires a human. An agent turn is a chain of model calls separated by tool executions, and any tool that runs longer than the marker TTL expires the cache *mid-turn*: build systems and test suites (2–20 min), media-generation polling (5–10 min per asset), rate-limit backoff, CI and deployment waits, sub-agent delegation, and approval gates all routinely exceed five minutes with no user absent. On a 5 m-TTL harness, every such tool forces a full-context re-write at 1.25× for the *next* model call in the same turn.

The per-event cost scales with working-context size. At a realistic agentic context of 100k tokens (Sonnet 4.5 list, $3/M input), one post-expiry re-write bills 100k × $3/M × 1.25 ≈ $0.38. An autonomous workload executing 30 long-running tools per day on a 5 m-TTL harness therefore pays ≈$11/day — ≈$250/month per seat — in cache-expiry tax that a 1-hour harness never incurs, *in addition to* the human-pause modeling above, and growing linearly with context length. The exposure is thus largest precisely where agent harnesses do their most valuable work: long-context, tool-heavy autonomous sessions. We note the inversion: the two harnesses marketed most explicitly for autonomous multi-step operation (OpenClaw, Hermes) are the two that selected the TTL least compatible with it.

8 Ancillary Finding: Subscription-Quota Exclusion

During configuration we observed Anthropic reject a third-party harness presenting a valid Claude-subscription OAuth token: *"Third-party apps now draw from your extra usage, not your plan limits. Add more at claude.ai/settings/usage."* (HTTP 400, `invalid_request_error`; raw response preserved in artifacts.) Third-party harnesses targeting Anthropic must therefore carry metered API credit irrespective of any subscription, while first-party-path harnesses retain subscription billing — an economic asymmetry layered on top of the caching results.

9 Threats to Validity

- **Single trial per cell.** Anthropic TTL expiry is contractual and the observed partition follows it exactly, but the Codex survival result is one sample of a load-dependent, unguaranteed mechanism.
- **Single idle duration.** 390 s cleanly separates the two TTLs; gaps > 1 h would defeat every harness tested absent keep-warm traffic.
- **Synthetic payloads.** Deterministic filler standing in for real work. Real sessions grow faster (tool results) and pause more often; both effects increase the measured gaps' magnitude, not their direction.
- **Footprint heterogeneity.** Absolute token counts are incomparable across harnesses by construction; all cross-harness claims use ratios, payload-normalized overhead, or the identical-content replay (§4.4).
- **Extrapolation assumptions.** Fixed break schedule and payload size; the model is linear in measured per-turn parameters and stated in full (§6) so readers may re-parameterize.
- **Monthly extrapolation.** §7 composes single-session measurements linearly across fixed daily profiles; real months vary payload size, schedule, and break structure. The direction of every stated bias (more frequent real-world pauses, growing Hermes leak, larger real payloads) widens rather than narrows the reported gaps, so the per-seat tax figures are conservative for the 5 m-TTL harnesses.
- **Version pinning.** Results describe the versions in Table 1 as of 2026-07-06/07; harness caching behavior can change across releases.

10 Reproducibility

All measurements derive from the published artifact set (browse · checksums):

```
results-e4b-agnt-v2.json          AGNT v0.6.4, Anthropic usage blocks, 5 turns + idle
results-e1-claude-code.json       Claude Code CLI JSON telemetry (session c43bb96a…)
results-e2-codex.json             Codex session-log last_token_usage extraction
results-oc-real.jsonl             OpenClaw embedded-agent lastCallUsage (Anthropic direct)
results-hermes-anthropic-direct.json  Hermes turns (proxy-verified)
results-omp-default.jsonl / results-omp-1hbest.jsonl   OMP both TTL paths (agent_end usage)
anth-proxy-log.jsonl / proxy-log.jsonl  Per-call markers + usage (ground truth)
results-e4-strategies.json        Controlled identical-content replay arms
e1-*.js e2-*.js e4*-*.js oc-turn.js anth-proxy.js hermes-anth-*.py   Runner scripts
```

Replication procedure: (1) install the six harnesses at the pinned versions; (2) start the logging proxy (`node anth-proxy.js`, listens on `127.0.0.1:18081`, forwards to `api.anthropic.com`, appends one JSONL line per `/v1/messages` call); (3) point OpenClaw/Hermes at Anthropic with a funded API key (Hermes: set `ANTHROPIC_TOKEN` — it takes precedence in its credential-resolution order — and pass `base_url` to `AIAgent`); (4) run four turns of Listing-1 payloads, sleep 390 s, run turn 5; (5) confirm guards G1–G3 held; (6) read the counters. Every number in Tables 2–4 is either a raw counter from these files or an arithmetic combination shown in the text.

11 Conclusion

Prompt-cache efficiency in deployed agent harnesses is determined by client-side engineering, and shipped behavior diverges sharply from what documentation implies. A single configuration bit — cache TTL — separates harnesses that resume an interrupted session at 10% of list price from harnesses that silently re-

purchase their entire context after every human-scale pause. Layered on top, we measure two independent cost pathologies: per-turn scaffolding injection that taxes every message at 2.3× payload (Claude Code), and marker placement that leaks premium writes even while the cache is alive (Hermes). OMP demonstrates that criterion (ii) is achievable independently of the others, yet it still ships the 5-minute default that forfeits (i). Of the six systems measured, only AGNT v0.6.4 simultaneously (i) survives realistic pauses via default 1-hour markers, (ii) bills premium tokens equal to the user payload alone, and (iii) does so across multiple providers — yielding the lowest measured and projected cost at every session length, with a cost multiplier that is uniquely monotone-decreasing in session duration. Composed over a working month, these per-session mechanisms compound into a per-seat cache tax of $205–$290/month ($2,500–$3,500/year) for the alternative Anthropic-path harnesses, and ≈$17,400/year for a five-seat team in the worst measured case — recurring spend separable from model quality entirely, and recoverable by client-side engineering alone.

References

1. Anthropic. *Prompt caching*. Claude Platform Documentation. platform.claude.com/docs/en/build-with-claude/prompt-caching (accessed 2026-07-06). Pricing multipliers: cache read 0.1×; 5-minute write 1.25×; 1-hour write 2× under beta header `extended-cache-ttl-2025-04-11`.
2. OpenAI. *Prompt caching*. OpenAI API Documentation. developers.openai.com/api/docs/guides/prompt-caching (accessed 2026-07-06). Automatic prefix caching; no retention control; load-dependent eviction.
3. M. Bolin. *Unrolling the Codex agent loop*. OpenAI Engineering, Jan 23 2026. openai.com/index/unrolling-the-codex-agent-loop.
4. Nous Research. *Context Compression and Caching*. Hermes Agent Developer Guide. hermes-agent.nousresearch.com/docs/developer-guide/context-compression-and-caching (accessed 2026-07-06).
5. OpenClaw. *Prompt caching*. Technical reference. docs.openclaw.ai/reference/prompt-caching (accessed 2026-07-06). `cacheRetention: none|short|long`; "short" (5 m) seeded by default for Anthropic.
6. Anthropic. Third-party subscription-quota policy change; error string captured live 2026-07-06, preserved in benchmark artifacts.

Technical Report v2.0 · AGNT Labs · Benchmark executed 2026-07-06/07 (OMP added 2026-07-07) · Model: claude-sonnet-4-5-20250929 (funded API key) + Codex default backend · Idle 390 s all arms · Payload ≈5,186 tokens/turn all arms · Correspondence: AGNT project. · Artifacts & raw counters