

# Cost logger over-count fix (cumulative re-sum) — MC #103671

## Cost logger over-count fix (cumulative re-sum ? idempotent per-session)

**Book:** System Architecture **Status:** Fixed + verified — MC #103671 (2026-06-15) **Commit:**  
`ae045e589` (~/.claude)

---

### The bug

`~/.claude/hooks/claude-cli-cost-hook.sh` is a **Stop hook**. Every time it fires (end of each turn) it parses the **entire** session transcript and sums `input_tokens + cache_creation` across **all** assistant messages, then **INSERTed a fresh** `cost_events` row with that cumulative total.

Because the transcript grows each turn, every firing logged an ever-larger cumulative snapshot of the *same* session. Across a day one session produced dozens of rows, so `SUM(cost_usd)` counted the same early tokens repeatedly.

### Evidence (tool-verified, costs.db)

- Today Opus `SUM` = **\$14,686** (129 events) vs `MAX` single = **\$478**.
- One event logged **6,959,199 input tokens** — physically impossible (context max 1M) → proves cumulative re-sum.
- All-time: 180 events >1.5M input tokens.

### Impact

1. **Killswitch** / `userprompt-cost-guard.sh` read `SUM(cost_usd)` for today → fired on phantom spend. Enabling the guard would have blocked **every** prompt. (Likely why the guard was previously removed — wrong fix.)
2. `cost-tracker.js` SUM-based reporting inflated ~30×.

## The fix

Before INSERT, **DELETE any prior row for the same** `session_id` (read from `metadata.session_id`, scoped to `source='claude-cli'`), so each session contributes exactly **one** row — the latest cumulative. `'unknown'` sessions skip the replace (avoid collapsing distinct parse-failures). No schema change.

```
if session_id and session_id != 'unknown':
    DELETE FROM cost_events
    WHERE source='claude-cli' AND json_extract(metadata,'$.session_id') = ?
INSERT ...
```

## Verification

- Hook run **3x** on a fixed transcript → **1 row** (was 3), cost `$0.1425` (exact: 3000 in / 1300 out @ opus 15/75 per-1M).
- One-time historical dedupe (keep max-cost row per session): claude-cli rows **4060 → 287** (= distinct sessions); today Opus SUM **\$10,997 → \$1,437**. costs.db backed up pre-dedupe; `PRAGMA integrity_check` = ok.

## Important follow-on (not a bug)

After correction, today's **real** Opus spend ≈ **\$1,437** — still 3× the `$500` daily ceiling and above the `$1000` killswitch. So there is a genuine cost signal, not pure phantom. Decision needed: raise the ceiling to reflect Opus-1M pricing reality, or treat as overspend. `userprompt-cost-guard.sh` restoration (MC #103654) stays paused until that ceiling decision, else it legitimately blocks.

---

Revision #1

Created 2026-06-15 15:40:56 UTC by John

Updated 2026-06-15 15:40:56 UTC by John