P4-CACHE

p4-cache

Tiered storage for Perforce — local-disk speed, object-storage prices.

Independently reviewed. Failure-tested. Reversible.

Cut your Perforce storage bill by ~85–90% — without changing a single developer workflow, and with `p4 sync` and `p4 submit` staying within ~10% of plain `p4d` (most configs within ~5%).

The problem: you pay premium-storage prices for cold data

Every Perforce shop above a few terabytes pays the same tax. Only 5–15% of a depot is read in any given month, but the *entire* depot sits on premium enterprise storage — because `p4d` doesn't distinguish the blocks hit constantly from the ones untouched since 2019. You pay the worst-case access price across 100% of the depot.

Enterprise arrays (NetApp / Pure / Isilon) run **$1,500–3,000/TB/year** fully loaded, and demand a seven-figure refresh every 4–5 years. Cloud premium SSD scales linearly with depot growth. Either way, the cold 85–95% of your depot is the most expensive data you own.

The usual workarounds each cost something: `+X` archive filetypes push per-file decisions onto the admin; `p4 obliterate` trades storage cost for irreversible deletion; hand-rolled s3fs/blobfuse setups work until the engineer who built them leaves or a quiet hash mismatch corrupts a sync six months later.

What p4-cache does

p4-cache keeps your hot working set on fast local NVMe and tiers the cold majority to cheap cloud object storage (S3 / Azure / GCS) or NFS. It sits transparently in front of your existing `p4d`:

- **No depot migration, no depot-format change, no client software.** Perforce is unchanged — no patches, no protocol changes, nothing developer-visible.
- **Cold files rehydrate on demand.** When `p4d` reads a tiered file, p4-cache fetches it from the backend, verifies it, and serves it — automatically.
- **The cloud is the source of truth.** Your durable copy lives in object storage; local NVMe is a fast cache in front of it.

You retire the expensive array and pay object-storage prices for the cold bulk, while developers keep local-disk speed.

It will not slow your developers down

This is the question every evaluation asks. The answer, measured on a re-runnable in-repo harness (500 files / 493 MiB, 32 concurrent readers, cache on local NVMe, across S3 / Azure / GCS / NFS):

- `p4 submit` **stays at native speed** — within **+1.3% to +3.5%** of plain `p4d` on every cloud backend. Submit returns as soon as archives hit the local cache; the durable cloud upload runs asynchronously behind it, so a commit is never gated on cloud latency.
- **Warm reads are local-disk speed** — **342–383 MB/s** against a **379 MB/s** plain-`p4d` baseline (within ~10%, most configs ~5%). Once a file is cached, `p4d` serves it locally with zero backend round-trips (regardless of packing); every config passed at 100% local-cache hit (zero re-fetches). This is the steady-state developer experience.
- **A cold miss pays the cloud once, then never again.** The first read of a tiered file after eviction costs one cloud round-trip, then it is cached. That is the one-time worst case, not steady state.

On network / NAS storage today? p4-cache can make reads *faster*. Moving your hot working set off a network filesystem onto local NVMe is architecturally faster than reading over the wire. (Our benchmark uses a local-directory stand-in for NFS, so it demonstrates no-regression parity rather than a measured network speedup — but the direction is structural.)

What you save

The license is a small per-capacity fee — **$100 per TB per year** — set deliberately below the storage savings it unlocks. You retire $1,500–3,000/TB array cost and replace it with cheap cold object storage ($24–150/TB/yr), commodity NVMe for the hot slice (~$0–300/TB/yr), and the $100/TB license.

Worked examples (≈10% hot on NVMe, ≈90% cold on object; array at $2,000/TB, object at $75/TB, NVMe at $200/TB):

Depot	On enterprise array	With p4-cache (hot + cold + license)	Annual saving
100 TB	$200,000	$18,750	**$181,250 · ~91%**
500 TB	$1,000,000	$93,750	**$906,250 · ~91%**
1 PB	$2,000,000	$187,500	**$1,812,500 · ~91%**

Cloud studio (50 TB on premium SSD @ $1,500/TB): $75,000/yr today → ~$9,875/yr with p4-cache = **~$65,000 saved (~87%)**. The $5,000 license is under 8% of the saving.

The array refresh you avoid — the seven-figure CapEx pulse every 4–5 years — is on top.

Proven, safe, and tested

The pillar most year-1 products can't claim.

- **Independent, adversarial code + security review.** A deep multi-subsystem scan surfaced **31 issues** — logic, failure-handling, and security findings across the daemon, shim, storage, and licensing — each adversarially verified and fixed.
- **Disaster recovery, proven.** The cloud object store is the source of truth. If the local database is lost, a fresh install rebuilds its catalog from the cloud (documented rebuild-from-footers recovery). Durability gaps were closed so a crash or loss never strands or deletes recoverable data.
- **Failure-injection tested (~45 tests) — no data loss, no corruption.** Recovery is proven across process crash / SIGKILL, Perforce-server outage, network / backend loss, disk-full / read-only, and bit-level corruption. Every corrupted byte-path is refused, never served.
- **Concurrency & recovery races, tested deterministically.** The hardest-to-catch failures — timing- and startup-order-dependent — are pinned by tests that force the exact interleaving: the upload pipeline is gated so crash-recovery can't clobber a live upload; multiple daemons over one bucket converge with no split-brain (tested at three writers); a crash mid-pack recovers on the next boot; and a restore stays correct while eviction runs concurrently. Part of a **~870-test** automated suite that grows with every hardening pass.
- **Content integrity, end to end.** Every restored file is **BLAKE3-verified** before it reaches `p4d` — on both the primary and the secondary-failover path — so a tampered or corrupted file fails the restore, never the depot. A checkpoint-verify tool offers three cost-tiered integrity levels.
- **Engineering rigor as a trust signal.** Every change must pass a strict quality gate: an 85% test-coverage floor, a zero-warnings lint wall, a minimum-supported-compiler build, fuzzing, a performance-regression gate, and dependency vulnerability / license auditing. Release binaries ship with build-provenance attestation and checksums.

> **For your security review:** the Security & Compliance Brief documents the threat model and enumerates defenses by category, so your security team gets the answers up front rather than weeks of investigation. Available under NDA.

Pricing

Flat per-capacity license. No per-seat fees, no feature gating. A floating trial is built in.

Depot size	Annual license
≤10 TB	**$1,000** *(floor)*
25 TB	**$2,500**
50 TB	**$5,000**
100 TB	**$10,000**
250 TB	**$25,000**

Depot size	Annual license
500 TB	**$50,000**
750 TB	**$75,000**
1 PB (1,000 TB)	**$100,000** *(cap)*

Above 1 PB is a custom ELA. **Commercial models:** Design Partner (first customers, 50% off year 1) · Annual (standard) · Perpetual (3× annual + 18%/yr maintenance) · ELA (1 PB+, custom).

Next steps

1. **Run the numbers** against your actual storage baseline (cloud, on-prem, or array-refresh quote).
2. **Request the briefs** for your technical and security teams: Architecture Deep-Dive, Security & Compliance Brief, Engineering Maturity Brief. Available under NDA.
3. **Scope a proof-of-concept** — measure on your hardware, your access patterns, your real depot.

Calculate your savings · Request an evaluation · Book a technical call

info@p4-cache.com *p4-cache.com*