Financially Optimal Expansion of a GPU Cluster Under a Given Workload: A Load-Curve Approach Applied to Small-Office Research Agents

Yash Singh

Developer and Mathematician

Iron.ws

iron.ws@proton.me

November 25, 2026

Abstract

A data center that serves small and medium-sized offices running LLM-based research agents must decide how many GPUs to own and when to add them, knowing that tenant demand is bursty, concentrated in working hours, grows in steps as offices are onboarded, and can always be overflowed to rented GPUs. We formalise the problem on the *load-duration curve*. For a convex cost model with amortised capital, idle and dynamic power, and a cloud overflow price, the optimal owned capacity is a *critical fractile* of the load distribution: $P(L > K^*) = c_f/(p_c - e)$. A first-stage simulation (mean load 4.4, peak 29.0 RTX 3090-equivalents) yields an optimum of 4 owned units, saving 18.4% versus all-cloud and 59.9% versus sizing to the 99.9th-percentile load. The recommendation is sensitive to the rental price: with all other assumptions unchanged, a price of \$0.20–0.22 per hour, which lies within the range listed by marketplaces in November 2026, removes the case for expansion. We then extend the model to a multi-tenant provider: seat-priced contracts, tenant onboarding and churn, a deferrable share of work that can be queued off-hours, and separate research and verifier model pools on 80 GB cards. In the reported run of 1,000 simulated 36-month paths, a load-curve-aware purchase rule raised mean NPV over renting-only by about \$159k (H100 case) or \$104k (memory-bound, A100 case), whereas a flat-demand rule lost money in 16.6–57.1% of paths. The deferrable share of work, fixed operating cost and rental prices dominate the outcome. The first-stage model is reproducible end to end from the supplied script. The multi-tenant simulator is part of the supplied script and is re-run by it unless the `--cached` option reuses a previous log. The multi-tenant results use heuristic purchase rules, split owned 80 GB cards between model pools in whole cards (a heuristic split that maximises large work served, not total cost), price power per card type and pool, spread deferrable work uniformly over the week, count terminal resale value in NPV, and assume 80 GB rents above the Vast.ai marketplace prices listed in November 2026. They are illustrative rather than validated financial forecasts.

1 Introduction

Research agents (code-writing, literature-review, evaluation and data-wrangling agents) are used by a handful of people per office, but each agent session may hold a model on one or more GPUs for an hour or more, and a second, independent model is often used to verify the first one's output. A data center that serves many small and medium offices sees the sum of these workloads: low on average, spiky during the working day, nearly empty at night, and growing in steps as new offices sign. Buying hardware for the peak wastes capital; renting everything pays a premium on every hour of the busy baseline. Because rental markets such as Vast.ai and RunPod expose GPUs at hourly prices [1, 8, 4], *the right question is not "buy or rent" but "where on the load curve should ownership stop and renting begin, and when should each purchase happen?"*

Contributions. (i) A closed-form critical-fractile rule for optimal owned capacity with cloud overflow, together with its exact integer-capacity counterpart; (ii) a stochastic workload generator and a quantified single-fleet analysis of savings, sensitivity, robustness and growth; (iii) a multi-tenant extension for a provider with seat-priced contracts, onboarding dynamics, deferrable work and split model pools, evaluated by Monte Carlo on purchase *rules*; (iv) a pipeline that regenerates the first-stage results, figures and this manuscript; the multi-tenant simulator is included in the same script and is re-run by it, or its previous log is reused with `--cached` (Section 6).

2 Background and Related Material

Rental prices. Marketplace and aggregator pages report hourly RTX 3090 prices spanning roughly an order of magnitude depending on provider tier and interruptibility [1, 2, 4, 5, 6, 7, 8], and survey articles compare RunPod, Vast.ai and Lambda [9, 10, 11]. Prices drift over time [3], which motivates our price-decline scenario.

On-premises hardware. For datacentre-class nodes, NVIDIA documents a DGX H100 with eight H100 GPUs and a maximum system power of about 10.2 kW [12, 13, 15], which is a substantial electrical and cooling requirement that a data center can plan for but an office cannot meet casually. We nevertheless treat H100-class capacity as a separate option in the first stage and as a first-class option in the multi-tenant extension.

Capacity planning. Our fractile result is the continuous analogue of the newsvendor problem and of classical "screening-curve" generation planning, where plant types are ordered by fixed and variable cost and each serves a slice of the load-duration curve. An additional reference on LLM serving in the supplied reading list is [18].

3 Problem Formulation

Let $L(t)$ be the load at time t in *3090-equivalent GPU units*, observed over a horizon of H hours. Own capacity K serves $\min(L, K)$; the excess $(L-K)^+$ is rented at price p_c per unit-hour (including a multiplicative overhead for storage, egress and setup). Owning one unit costs a fixed c_f per hour irrespective of use,

$$c_f = \frac{C\,[\mathrm{CRF}(r,n)+m]+o}{8760} + \kappa\,\phi\,\pi\,\mathrm{PUE},$$

where C is capex per unit, $\mathrm{CRF}(r,n) = r/(1-(1+r)^{-n})$ is the capital-recovery factor at discount rate r and life n years, m is the maintenance fraction, o the yearly administration cost, κ the unit's power in kW, ϕ its idle fraction and π the electricity price. Thus c_f is the sum of four hourly components: capital recovery $C\,\mathrm{CRF}(r,n)/8760$, maintenance $C\,m/8760$, administration $o/8760$ and idle electricity. Running a unit costs an additional dynamic energy $e = \kappa(1-\phi)\pi\,\mathrm{PUE}$ per busy unit-hour. Costs are accumulated over the simulated horizon of 8,736 h, written H (52 weeks) using hourly rates derived from the 8,760-h year, so the "annual" costs below are 52-week costs; the 0.3% difference does not affect any decision. The cost is

$$\mathcal{C}(K) = c_f\,K\,H + e\int_0^H \min(L,K)\,dt + p_c\int_0^H (L-K)^+ dt. \tag{1}$$

Proposition 1 (Critical fractile, continuous load)**.** *If $p_c > e$ and L has a continuous distribution, $\mathcal{C}$ is convex in K and minimised at K^* satisfying*

$$P(L > K^*) = \frac{c_f}{p_c - e} \equiv u^*, \tag{2}$$

with $K^* = 0$ if $u^* \geq 1$. If $p_c \leq e$, renting is never dearer than running an owned unit, so no capacity is worth owning ($K^* = 0$, or $K^* = K_0$ in the brownfield case).

Proof. $\mathcal{C}'(K)/H = c_f + e\,P(L>K) - p_c\,P(L>K)$. Since $P(L>K)$ is non-increasing, $\mathcal{C}'$ is non-decreasing, so $\mathcal{C}$ is convex; setting the derivative to zero gives the result. □

Proposition 2 (Integer capacity)**.** *Assume $p_c > e$ (otherwise $K^* = 0$, as in Proposition 1). For integer K,*

$$\frac{\mathcal{C}(K+1) - \mathcal{C}(K)}{H} = c_f - (p_c - e)\,\mathbb{E}\big[\min\big((L-K)^+, 1\big)\big]. \tag{3}$$

The right-hand side is non-decreasing in K, so the optimal integer capacity is the smallest K with $\mathbb{E}[\min((L-K)^+, 1)] \leq u^*$.

Proof. $\min(L, K+1) - \min(L,K) = \min((L-K)^+, 1)$ and $(L-K-1)^+ - (L-K)^+ = -\min((L-K)^+, 1)$; substitute in the cost and take expectations (time averages). □

If L takes only integer values, $\min((L-K)^+, 1) = \mathbf{1}\{L > K\}$ and Proposition 2 reduces to the familiar rule "smallest K with $P(L > K) \le u^*$". Our simulated load contains half-unit sessions, so it is not integer-valued, $\mathbb{E}[\min((L-K)^+, 1)]$ is slightly smaller than $P(L > K)$, and the two criteria can disagree in general. We therefore use Proposition 2 for integer decisions and Proposition 1 as the continuous benchmark. The quantity u^* is the *break-even utilisation*: a marginal owned unit pays off only if it is busy more than a fraction u^* of the time. Graphically, K^* is where the load-duration curve crosses height u^* on the time axis (Fig. 2). *Brownfield* case: if K_0 units already exist, only their capital-recovery charge is sunk; their maintenance, administration, idle and energy costs continue. We optimise over $K \ge K_0$ with capital recovery charged only on new units. The sunk charge is a constant for $K \ge K_0$, so it changes the level of reported costs but not the optimal K.

4 Data and Calibration

In Sections 4–5 the load $L(t)$ is the *aggregate* demand placed on the facility by a cohort of tenant offices, in 3090-equivalent units. Because tenants pay a fixed per-seat price, revenue does not depend on the fleet, so minimising cost and maximising profit coincide; Section 6 drops this simplification.

Workload. Agent sessions arrive as a non-homogeneous Poisson process (up to 5 sessions/h on weekdays 09–18, lower shoulders, a night/weekend floor), with lognormal durations (mean 1.5 h) and sizes in {0.5, 1, 2, 4} equivalent units; a nightly evaluation batch (6 units, 3 h) runs with probability 0.7. This is an *assumed* generator, not measured traffic; replace it with logs from your own scheduler for a real decision. Simulated 52-week load: mean 4.4, median 2.5, p90 11.5, p99 19.0, max 29.0 units (load factor 15%), i.e. 38,825 unit-hours per year.

Costs. Table 1 lists derived per-unit costs. Baseline inputs for the 3090 class: capex \$1,300 per card including host share, life 3 years, cloud price \$0.25/h, electricity \$0.13/kWh, PUE 1.25, discount rate 8%, maintenance 5%/yr, cloud overhead ×1.10. For H100 we use \$35,000 per GPU (DGX share), 1.275 kW per GPU (an allocated share of the 10.2 kW system maximum divided by eight, not a measured per-GPU draw), \$2.50/h cloud and one H100 ≈3.5 3090-equivalents on LLM inference. **All of these are planning assumptions placed within the ranges reported by the cited sources; none was scraped, and they should be replaced with current quotes in `JSON/params.json`.** Section 5.5 compares the 3090 rental assumption with prices listed in November 2026.

Table 1: Per-unit economics (\$ per 3090-equivalent unit-hour).

Class	c_f	e	p_c	u^*
RTX 3090 (24 GB)	0.093	0.062	0.275	44%
H100 (80 GB, DGX node share)	0.462	0.047	0.786	63%

5 Results

5.1 Load curve and optimal capacity

For the 3090 class $c_f = \$0.093$, $e = \$0.062$, $p_c = \$0.275$, hence $u^* = 44\%$ (fractile 43.7% of the year). The load-duration curve (Fig. 2) gives an analytic greenfield capacity of 4 units; the numerical optimum is 4 (greenfield) and 4 starting from the existing $K_0 = 2$ (Fig. 2). The exact integer condition of Proposition 2 agrees: going from 3 to 4 owned units lowers cost by \$50.64 per 52-week year, whereas going from 4 to 5 would raise it by \$47.06. At the optimum the owned units run at 57% average utilisation, the cluster is saturated and bursting to cloud for 42% of the year, and 49% of unit-hours are rented.

5.2 Strategy comparison

The optimal expansion saves 18.4% over renting everything, 2.1% over not expanding, and 59.9% over sizing for the 99.9th percentile.

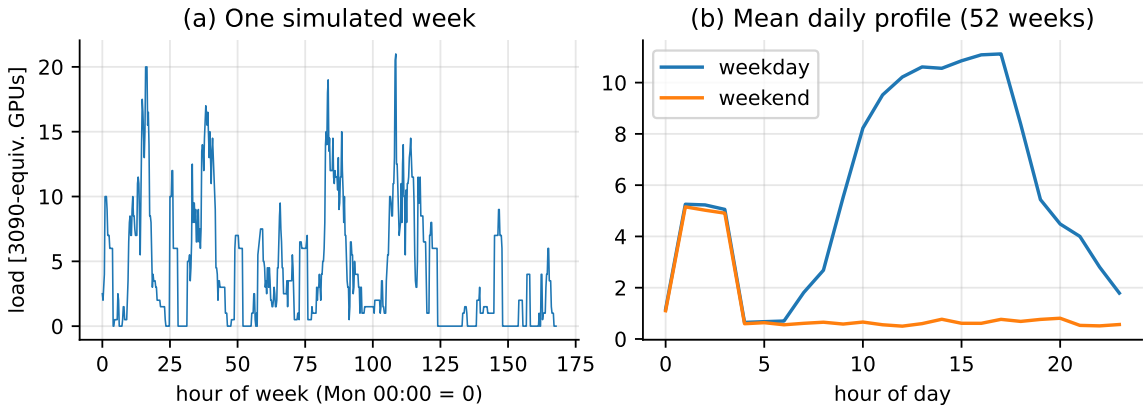

Figure 1: Simulated agent workload.

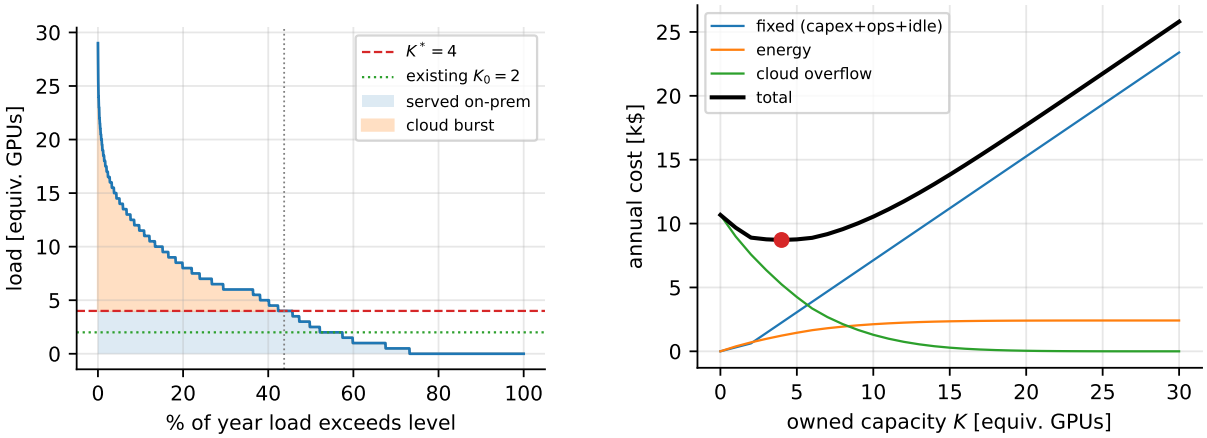

Figure 2: Left: load-duration curve; ownership covers the lower slice, cloud the peak. Right: cost components versus owned capacity.

5.3 Break-even and hardware class

Per 3090-equivalent, H100 capacity has c_f = \$0.46/h and p_c = \$0.79/h, with break-even utilisation 63%. A single DGX-style node of eight H100s provides 28 equivalents, costs about \$280,000 and draws up to 10.2 kW, with roughly \$113,442 of fixed annual cost. On raw throughput-per-dollar the 3090 class wins at this workload, but its 24 GB limits model size; if a meaningful share of sessions needs 70B-class models at higher precision, those sessions should be modelled as a second load curve served by rented H100-class GPUs or by an H100 node only if that curve itself has a high duty cycle. Our model, applied to each curve separately, gives this decision directly.

5.4 Sensitivity

5.5 Sensitivity to observed rental prices

The recommendation is driven by the rental price p_c. Table 4 re-optimises with only the base 3090 cloud price changed (the $\times 1.10$ overhead is still applied); workload, electricity, ownership and overhead assumptions are as in the baseline. Pages read in mid-November 2026 list RTX 3090 offers on Vast.ai from about \$0.09 to \$0.11 per hour with a median near \$0.20 [16], and RunPod lists a community price of \$0.22 and an on-demand price of \$0.50 [17, 8, 6]; community-tier listings depend on stock and may not be available at the listed price, so the \$0.22 figure should be read as a lower bound on what can actually be rented. With unchanged assumptions the baseline four-unit expansion is no longer justified at \$0.20–0.22 per hour: the brownfield optimum stays at the two existing cards, and expansion first pays at a base price of about \$0.225 per hour. At the lowest listed prices the break-even utilisation exceeds 100%, so a greenfield fleet should not be owned at all. The \$0.25 baseline therefore sits above the listed marketplace and community prices

Table 2: 52-week cost ($) by strategy, 3090 class. Existing units carry maintenance, administration, idle and energy costs but no capital recovery.

Strategy	Fixed	Energy	Cloud	Total
All cloud (K=0)	0	0	10,677	10,677
Keep K0=2 + burst	620	700	7,579	8,900
Optimal expansion (K=4)	2,247	1,229	5,240	8,716
Peak-sized (K=25, p99.9)	19,326	2,413	3	21,741

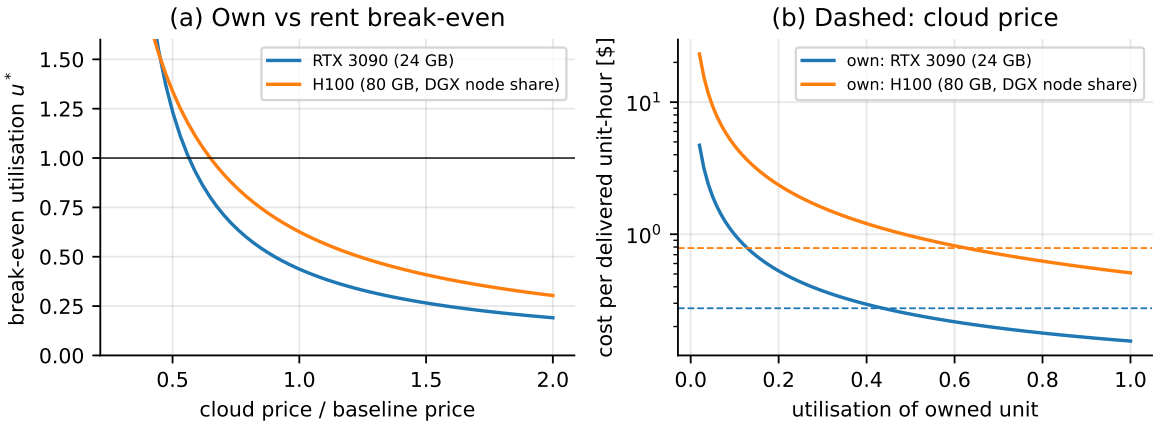

Figure 3: (a) Break-even utilisation versus cloud price. (b) Cost per delivered unit-hour for an owned unit versus utilisation; dashed lines are cloud prices.

and well below the listed on-demand price of $0.50: expansion is supported if the relevant alternative is something like the baseline, and is not supported if it is cheap interruptible marketplace capacity. Which tier is the relevant opportunity cost, with its interruptions, reliability and confidentiality differences, is a decision input the model cannot supply. The same caution applies to the 80 GB class: Vast.ai's H100 SXM listings on 16 November 2026 had a median of about $1.93 per hour (cheapest $1.73) [16], below the $2.50 baseline, while other providers list higher prices by tier and form factor.

5.6 Robustness to workload realisation

Across 60 simulated years the optimal capacity ranges from 4 to 4 (mean 4.0). Using one year's optimum on all others incurs a mean regret of 0.0% (max 0.0%), because the cost curve is flat near its minimum (Fig. 2); mis-sizing by one unit is cheap compared with choosing the wrong regime.

5.7 Growth and falling cloud prices

With workload growing 35%/yr, the capacity path and costs are in Table 5 and Fig. 7; the second K column assumes cloud prices fall 15%/yr. Falling rental prices delay or shrink purchases: the option value of waiting is real, and the fractile rule should be re-run each planning cycle with current quotes. Cost accounting: each year's cost charges capital recovery and maintenance on every owned card beyond the K_0 existing ones, so cards bought earlier stay on the books. Because capital already committed is a constant in the choice of $K_t \geq K_{t-1}$, the capacity path is the same as it would be if earlier purchases were treated as sunk, but the reported costs are higher. The table is a sequence of steady-state annuity comparisons, not a discounted cash-flow model: it does not track purchase dates, upfront outlays, resale values or the replacement of cards whose three-year life ends inside the five-year horizon, so the yearly savings should not be read as a validated five-year investment return. A cash-flow version of this analysis is left for future work.

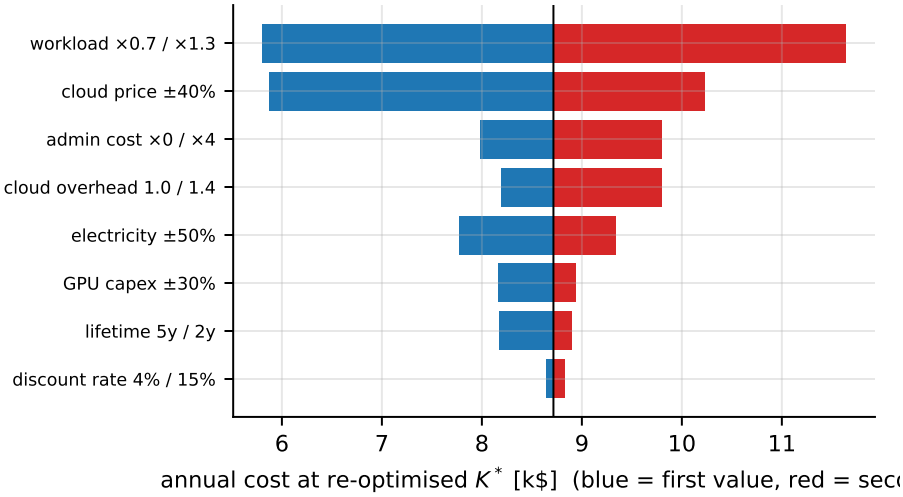

Figure 4: One-at-a-time sensitivity of optimal annual cost (re-optimised K).

Table 3: Optimal K and annual cost under low / high parameter values.

Parameter	K^* (low / high)	Cost $ (low / high)
workload ×0.7 / ×1.3	3 / 5	5,805 / 11,639
cloud price ±40%	2 / 6	5,868 / 10,226
admin cost ×0 / ×4	6 / 2	7,986 / 9,797
cloud overhead 1.0 / 1.4	3 / 6	8,188 / 9,799
electricity ±50%	6 / 2	7,769 / 9,346
GPU capex ±30%	6 / 2	8,163 / 8,938
lifetime 5y / 2y	6 / 2	8,170 / 8,900
discount rate 4% / 15%	4 / 3	8,644 / 8,831

6 Multi-Tenant Data Center Extension

The first-stage model treats demand as a stationary load. A provider's demand is instead built bottom-up from tenants, and its decision is dynamic. We therefore incorporate an additional simulator (formerly the file `office_agents_model.py`, now merged into the supplied `make_paper.py`). Run normally, the script executes this simulator, writes its log to `office_extension/office_run_log.txt`, parses that log into `office_results.json` and copies its figures into `out/`; every number in this section is parsed from that log. With `--cached` the simulation is skipped and a previous log or `office_results.json` is reused, so the numbers in a given manuscript are only as current as that log. Path counts are set by the environment variables `NMC` and `NSENS` (defaults 1000 and 200). The figures reported here come from one full-size run of this script (1,000 Monte Carlo paths per scenario, 200 per sensitivity case). The main Monte Carlo uses seed 7 and the sensitivity cases use seed 11; the log header records both seeds, the path counts and a hash of the script (dc297222ce3e123a). A code review also showed several modelling choices that qualify its conclusions; they are stated where they apply and collected in Section 8.

6.1 Model

- **Bottom-up demand.** An office has 50 seats, 70% active, 8 research runs per active user per workday, 21.7 workdays, 120 H100-seconds per run, and a verifier consuming as much GPU time as the research model (×2 in total). This gives 405 H100-hours of 80 GB-class work plus 400 3090-hours of small-model work (claim checks, embeddings, extraction) per full office-month. Per-run GPU seconds are an assumption.

- **Load-duration allocation.** A 168-hour weekly profile follows office hours for the non-deferrable share and is flat for the deferrable share (queued deep research and verification). With deferrable share d and normalised business-hours profile b, weekly demand is $w = (1 - d)\, b + d$: the deferrable work is spread uniformly over all 168 hours, *including business hours*, rather than scheduled into

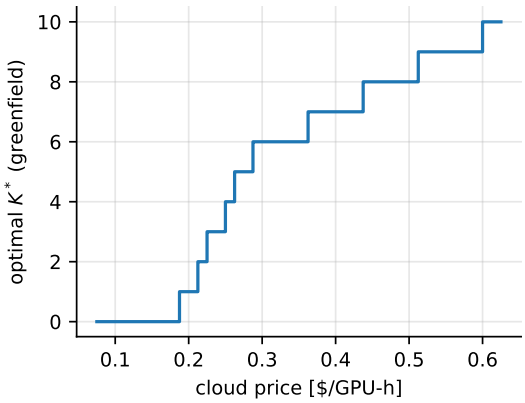

Figure 5: Optimal greenfield capacity as a function of cloud price: ownership is the dominant strategy only while rental is expensive.

Table 4: Optimal 3090 capacity and cost at different base rental prices (52-week costs in $; savings versus renting everything and versus keeping the $K_0 = 2$ existing cards).

Base price $/h	u^*	K^* (green)	K^* (from K_0)	Cost at optimum	vs all-cloud	vs keep K_0
0.11	>100%	0	2	4,655	0.9%	0.0%
0.20	59%	1	2	7,384	13.6%	0.0%
0.22	52%	2	2	7,990	15.0%	0.0%
0.25	44%	4	4	8,716	18.4%	2.1%
0.50	19%	8	8	11,693	45.2%	29.0%

off-peak slots subject to deadlines and queue limits. This amounts to perfect 24/7 smoothing, which raises the utilisation of owned cards and so favours ownership relative to a model with real deadlines; the results below are conditional on it. Owned cards serve demand hour by hour; the peak is rented, the base is owned (Fig. 8).

- **Separate pools.** Research and verifier models must be of different families, so owned 80 GB cards are split into two pools; merging them changes little (Table 10). Owned 80 GB cards are assigned to the two pools in whole cards, and the assignment (over all per-type card counts) is the one that serves the most large work, so in a mixed A100/H100 fleet the pools are balanced by capacity and not by card count. The physical side effect remains: a single owned 80 GB card serves one pool only and the other pool's work is rented, which understates the value of very small fleets, since a real card could serve either model at different times. Power is computed per card type and per pool from that pool's own mean utilisation, with idle 80 GB capacity shared out to small jobs in proportion to each pool's spare capacity; because card power is linear in utilisation this is exact for the mean. Electricity is the smaller cost here (about $128 per H100 per month at full load with the cost-module inputs of $0.28/kWh, PUE 1.25 and 0.50 kW). Merging the pools changes the mean gain little (Table 10).

- **Tenant dynamics.** Offices onboard as a Poisson process with lognormal signing rate (median 0.8/month, 2 pilots at month 0), adopt along a ramp (35%, 67.5%, 100%), vary in size, churn at 1.5%/month, and usage per seat drifts upward.

- **Contracts.** A fixed $50 per seat-month with no pass-through of rental price changes, so rental price risk sits with the provider. Renting from trusted providers (rather than marketplaces) can carry a premium.

- **Two hardware scenarios.** A: speed-ups of the earlier cost model (H100 is the efficient 80 GB card). B: memory-bound inference, where A100 and H100 speed-ups are 2.4× and 3.2× a 3090 (A100 becomes efficient).

Card prices, rents, power and availability come from the cost module of the earlier model and are listed in Table 6. Other inputs: 92% of hours usable, electricity $0.28/kWh with PUE 1.25, discount rate 12% per year, a 10% selling cost on resale, operating cost $250 per month once any hardware is owned plus

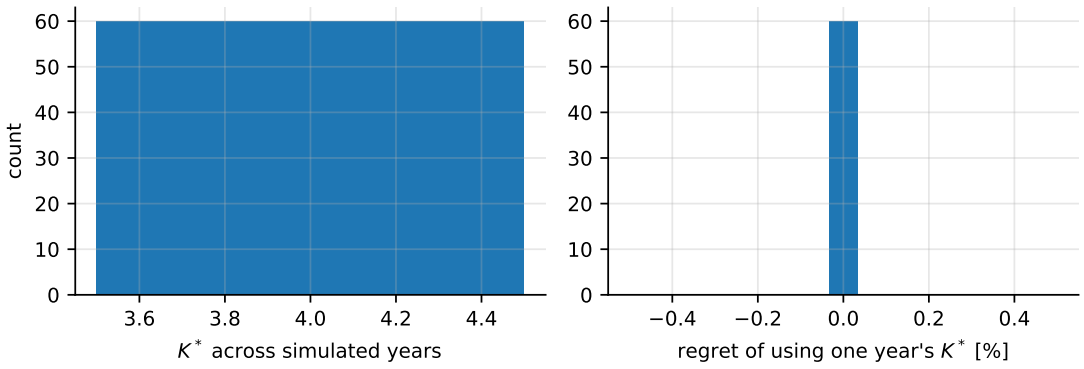

Figure 6: Distribution of optimal K and regret across simulated years.

Table 5: Growth plan (3090 class). K columns: flat price / declining price. Annuity-based 52-week costs in $ at flat prices; cards bought in earlier years keep their capital recovery and maintenance charges, only the K_0 existing cards are sunk.

Year	Load ×	K (flat)	K (decl.)	Optimal	All cloud	No expansion
0	1.00	4	4	8,716	10,677	8,900
1	1.35	5	4	12,133	14,414	12,546
2	1.82	7	4	16,722	19,459	17,498
3	2.46	10	4	22,917	26,269	24,250
4	3.32	13	4	31,295	35,463	33,399
5	4.48	18	4	42,588	47,875	45,775

$40 per owned card, starting cash $40,000 with a $3,000 reserve, and a 36-month horizon. Rents follow a monthly log-AR(1) index (persistence 0.90, volatility 0.06) that reverts towards 85% of today's level, so rents are expected to fall; the resale value of 80 GB cards scales with that index to the power 0.5. Purchased cards arrive one month after the order. These inputs differ from the first-stage ones (Section 8). Objective: NPV (36 months) of a purchase *rule* minus the NPV of renting only, over common random paths; the NPV includes the resale value of the remaining fleet at month 36 (after the selling cost).

6.2 Steady-state optimum

For a fixed number of fully adopted offices (20), a grid search over 3090/A100/H100 counts (3090 counts limited to 0–8) gives Table 7 and Fig. 9. At 40% deferrable work the provider should own 20 H100s in A (cost $18.9 per seat-month versus $29.1 renting, a saving of 35%), or 13 A100s in B ($16.2 versus $22.1, 27%). The saving rises almost linearly with the deferrable share, from 13% to 57% in A and 5% to 49% in B, because queued work flattens the load curve and raises utilisation of owned cards. The owned fleet moves in opposite directions in the two scenarios: it falls from 24 to 16 H100s in A and rises from 7 to 18 A100s in B. This is consistent with the fractile logic: ownership extends down the load-duration curve only while the marginal card is busy enough to beat renting. Reading Fig. 8(b), demand exceeds the 20-card optimum in roughly one fifth of hours, which is where the extension's marginal card stops paying. No 3090s appear in the steady-state optimum, but the search only tried 0–8 of them, so this shows that they were not optimal within that range, not that no 3090 fleet could be; with a 1.5× (small) / 1.3× (large) trusted-provider premium the optimum rises to 26 H100s (A, saving 47%) or 20 A100s plus 4 H100s (B, 36%).

6.3 Dynamic purchase rules

Real purchases happen under uncertain onboarding. The supplied study compares rules by Monte Carlo over 1,000 paths: renting only; a fixed roadmap; a gate that buys when projected rental savings exceed projected cost by a margin (15%), using either a flat-demand estimate or the full load-duration curve ("load-curve aware"); and gates restricted to A100, H100, both, or all three types. Tables 8 and 9 and Fig. 10 summarise the NPV gain. The gate is a heuristic with fixed design parameters: it forbids purchases before month 3, allows up to six cards in the first purchase and two in each later one, multiplies projected demand by a

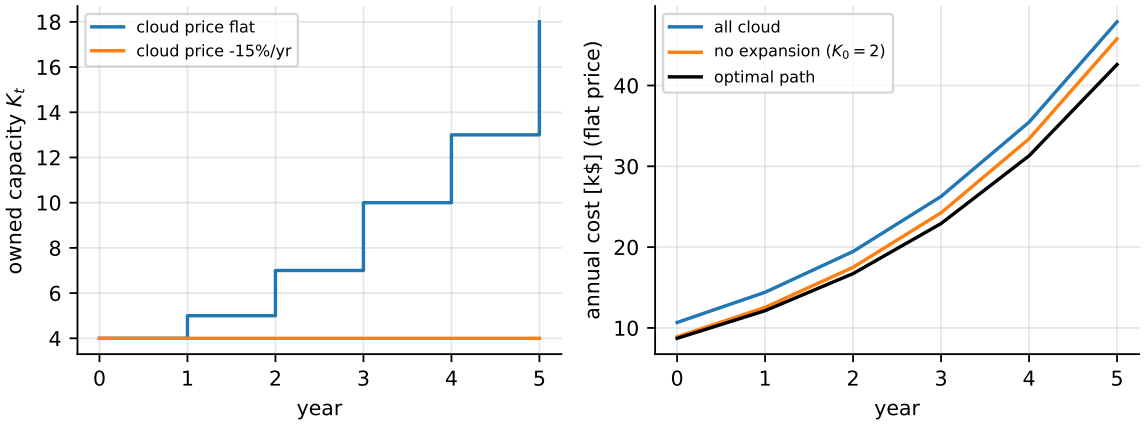

Figure 7: Expansion path and cost under growth.

Table 6: Hardware inputs of the multi-tenant cost module (scenario A speed-ups; scenario B uses 2.4× for the A100 and 3.2× for the H100). Price includes host share; rents are on-demand $ per GPU-hour; depreciation is the annual loss of resale value.

Card	Price + host $	Speed-up vs 3090	kW load / idle	Rent $/h	Depreciation
RTX 3090 (24 GB)	2,200	1.0	0.40 / 0.08	0.35	25%
A100 (80 GB)	16,000	3.0	0.45 / 0.10	1.79	15%
H100 (80 GB)	26,000	5.5	0.50 / 0.10	3.25	15%

factor of 1.8, and sells cards after month 12 when their utilisation is below 30%. These choices shape the results. In particular, the month-3 first purchase reported below is the earliest date the rule permits, not an optimum found by the simulation, and the sensitivity analysis varies the margin but none of the other gate parameters. The P(loss) column of the tables reports the raw number of paths, out of 1,000, in which the gated rule earned less than renting only (the log also prints the percentage to one decimal). A count of zero bounds the true loss probability by about $3/n$ at 95% confidence, i.e. 0.3% for the main runs. The main findings are:

- **Load-curve awareness matters more than hardware choice.** The flat-demand H100 rule over-buys (about 865k$ of capex in A and 953k$ in B), earns 119k$ and -2k$ on average, and loses against renting in 16.6% and 57.1% of paths. The same H100 gate with load-curve awareness earns 159k$ (A) and 81k$ (B) with loss probabilities of 0.3% and 2.5%, on less capex (573k$ versus 865k$ in A, 450k$ versus 953k$ in B).
- **Hardware fit.** In A, the load-aware H100 gate (159k$) beats the A100 gate (124k$); in B the order reverses (A100 104k$ versus H100 81k$). Letting the rule also choose 3090s adds nothing (on average 0.9 and 0.2 cards in A and B).
- **Fleet size.** The load-aware gate ends with about 22 H100s (A) or 22 A100s (B), i.e. 0.67 and 0.67 cards per office at the end of the horizon, with a first purchase around month 3 (the earliest month the gate allows) when about four offices are on board. The user roadmap (two 3090s in month 0, then utilisation-triggered purchases up to 4 H100s and 8 3090s) is safer but earns only about half as much.

Fig. 11 shows the growth path. In A the dynamic rule tracks average demand per office (0.60 H100) rather than the steady-state optimum, which is higher because ownership extends into the shoulder of the load curve; in B it stays below the average-demand line because the A100 only pays at high utilisation.

6.4 Sensitivity

Table 10 and Fig. 12 give one-at-a-time sensitivities of the mean NPV gain of the A100+H100 gate (baseline $163k in A and $109k in B at 200 paths; entries are k$ with the probability of losing against renting in brackets). Under the uniform-smoothing treatment of deferrable work, the deferrable share is decisive: with none of the work deferrable, owning gains nothing in A (mean $-2k, loss in 58% of paths) and is

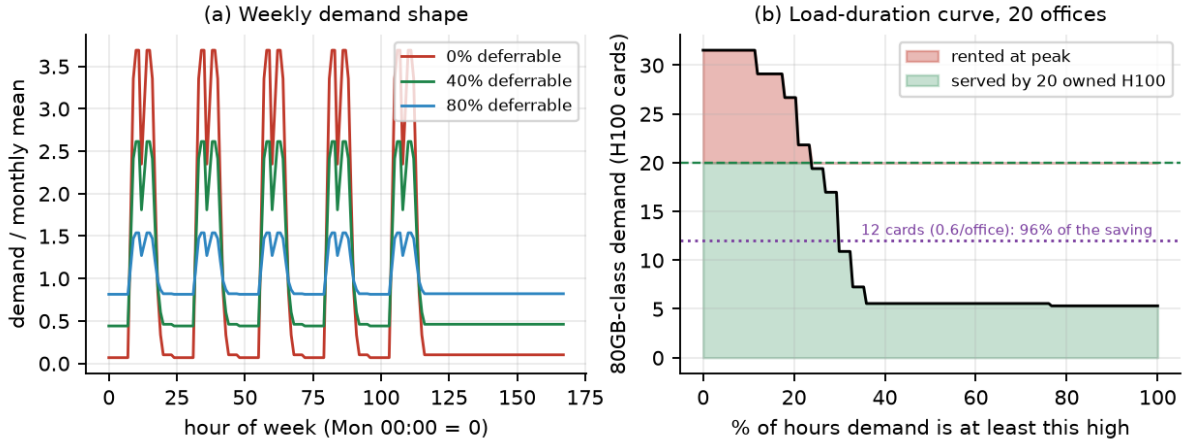

Figure 8: (a) Weekly demand shape for three deferrable shares. (b) Load-duration curve for 20 offices (40% deferrable, scenario A): 20 owned H100s serve the base while the peak is rented; the dotted line marks 12 cards (0.6 per office), which the script prices directly: it captures 96% of the saving.

Table 7: Steady-state optimum for 20 offices. Cost in $ per seat-month, own / rent-only.

Deferrable	Scenario A (H100)				Scenario B (A100)			
%	cards	saving	util.	cost	cards	saving	util.	cost
0	24	13%	35%	25.4 / 29.1	7	5%	53%	21.0 / 22.1
20	22	24%	44%	22.1 / 29.1	10	16%	72%	18.6 / 22.1
40	20	35%	54%	18.9 / 29.1	13	27%	83%	16.2 / 22.1
60	18	46%	66%	15.6 / 29.1	15	38%	93%	13.7 / 22.1
80	16	57%	81%	12.4 / 29.1	18	49%	97%	11.3 / 22.1

marginal in B; at 80% the gain is about $250k and $175k. Work volume per run (priced proportionally) scales the gain strongly. Rental price levels matter (80 GB rent −30% cuts the B gain from $109k to $23k with a 26% loss probability), card prices matter in the opposite direction, and fixed operating cost is a scale threshold: at $2,500 per month the loss probability rises to 10% (A) and 24% (B), which argues for pooling many tenants in one facility. The pool split, electricity price and gate margin are second-order. A rental premium for trusted providers, which providers may face for confidential workloads, *increases* the value of owning: ownership gives privacy and a hedge against rental price at once.

Rents versus listed prices. The cost module assumes on-demand rents of $1.79 per A100-hour and $3.25 per H100-hour. Vast.ai's marketplace listings on 16 November 2026 had medians of about $0.84 (A100 SXM4) and $1.93 (H100 SXM), with cheapest offers of $0.35 and $1.73 [16]. The A100 assumption is therefore more than twice the listed median and the H100 assumption roughly two thirds above it; the tested −30% range for 80 GB rents ends at $2.27 for the H100, still above the listed median, so the sensitivity analysis does not reach these prices. Since lower rents cut the gain sharply (Table 10), the reported gains probably overstate what marketplace prices would allow; reliable-tier prices may of course be higher than marketplace medians.

7 Worked Expansion Examples

This section turns the conclusions into concrete expansion plans for the data center. Every figure is computed by `make_paper.py` from `results.json`, `office_results.json` and `params.json`. Dollar amounts inherit the assumptions of Sections 4 and 6. "Revenue that justifies the expansion" is measured in two ways: the *margin uplift* (tenant revenue is fixed per seat, so owning raises the margin by lowering the cost to serve), and the *tenant revenue that triggers each purchase*. Monthly cost figures include the capital charge (amortisation) of owned cards, so a positive saving means the hardware has more than paid for its capital cost; the cash capex is shown separately.

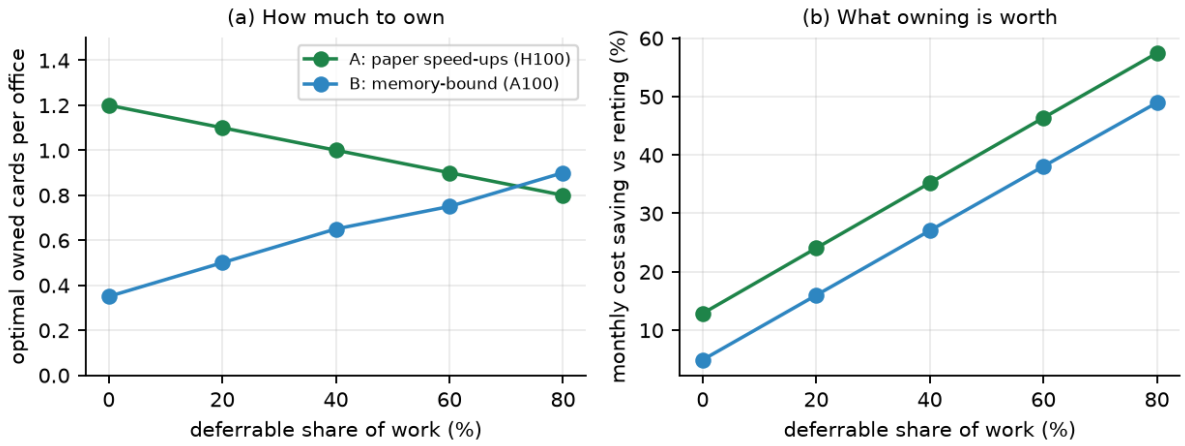

Figure 9: Optimal owned cards per office and saving versus renting, as a function of the deferrable share.

Table 8: Scenario A: NPV gain over renting only (k$, 36 months); losses = number of the 1,000 paths in which the rule earned less than renting only; fleet is H100/A100/3090 at the end.

Rule	mean	p5	median	p95	losses	capex	fleet
User roadmap (paper)	81	26	83	128	13	120	4.0/0.0/7.9
H100 gate, flat-demand rule (paper)	119	-47	94	368	166	865	33.3/0.0/0.0
H100 gate, load-curve aware	159	30	133	377	3	573	22.0/0.0/0.0
A100 gate, load-curve aware	124	31	109	265	0	383	0.0/24.0/0.0
80GB mix gate (A100+H100)	153	27	128	364	7	566	20.1/2.7/0.0
All-types gate (3090+A100+H100)	152	27	126	364	7	566	20.0/2.7/0.9

7.1 Example 1: starter cluster of 3090-class cards (Section 5 model)

Starting from the 2 existing cards, the plan in Table 11 buys one unit at a time as the load grows 35% a year, at an assumed $1,300 per card including host share. In year 0 the provider buys 2 cards ($2,600 cash) and thereby spends $184 less per year than by not expanding and $1,961 less than by renting everything. By year 5 the fleet reaches 18 units (cumulative expansion capex $20,800) and the annual avoided spend versus all-cloud is $5,287. At this scale the saving is modest in absolute terms, which is why the data-center view of Example 2 matters.

7.2 Example 2: data center with 20 tenant offices

Take 20 fully adopted offices of 50 seats at $50 per seat-month, so revenue is $50.0k per month ($2,500 per office), with 40% of the work deferrable. Card costs are the cost module's purchase price plus host share, $26.0k per H100 and $16.0k per A100 (Table 6); they equal the Monte Carlo runs' mean cumulative capex divided by mean terminal fleet ($26.0k and $16.0k), which suggests that the gated rules sold few or no cards. Table 12 compares three expansions against renting everything.

- **How to expand.** In scenario A buy H100s up to about 0.6 per office; owning them lifts the gross margin from 42% (renting everything) to 62% (full optimum) of revenue. In scenario B buy A100s instead, lifting the margin from 56% to 68%. Rent the remaining peak.
- **Where to stop.** Pricing the 12-card fleet directly in the cost model (20 offices, 40% deferrable, scenario A) gives 96% of the full-optimum saving; the ladder in Example 3 applies that share pro rata. On this basis the last 8 cards cost $208.0k and add only about $431 per month of net saving, so a gate that demands a 15% margin on projected savings would stop well short of 20.
- **Revenue that justifies it.** Each office adds $2,500 per month and, in A, about $488 per month of extra margin once the 12-card fleet serves it. Owning breaks even on cost per seat at $18.9 (A) or $16.2 (B) per seat-month, against $50 charged, which leaves room to cut prices to win tenants while staying profitable.

Table 9: Scenario B (memory-bound): same layout.

Rule	mean	p5	median	p95	losses	capex	fleet
User roadmap (paper)	41	1	41	77	46	121	4.0/0.0/7.9
H100 gate, flat-demand rule (paper)	-2	-145	-18	176	571	953	36.7/0.0/0.0
H100 gate, load-curve aware	81	8	64	204	25	450	17.3/0.0/0.0
A100 gate, load-curve aware	104	21	85	247	1	358	0.0/22.4/0.0
80GB mix gate (A100+H100)	104	21	85	247	1	358	0.0/22.4/0.0
All-types gate (3090+A100+H100)	103	21	85	247	1	359	0.0/22.4/0.2

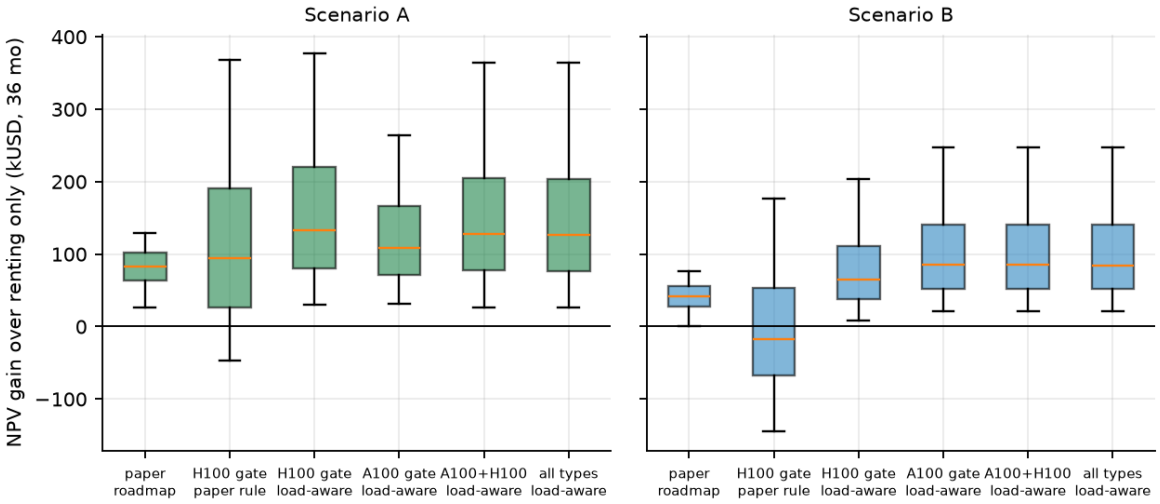

Figure 10: Distribution (5th–95th percentile whiskers) of NPV gain over renting only.

7.3 Example 3: staged purchases as offices are onboarded

Because tenants arrive in steps, purchases should follow them. The supplied gate buys when projected rental savings exceed the projected cost of the card by 15%, and first triggers around month 3 with about four offices on board (month 3 is also the earliest purchase date the gate permits). Table 13 applies the 0.6-card-per-office rule of scenario A, pro rata to the 20-office result (the supplied model reports the exact case only at 20 offices, so the intermediate rows are an extrapolation).

When the expansion is *not* justified. The same results show the conditions to avoid: no deferrable work (scenario A mean gain $-2k, loss in 58% of paths), a fixed operating cost of $2,500 per month with few tenants (loss in 10% / 24% of paths in A / B), or a 30% fall in 80 GB rental prices in the memory-bound case (gain falls from $109k to $23k). In the first-stage model, marketplace prices of $0.20–0.22 per hour for 3090-class capacity also remove the case for expansion (Section 5.5). Re-run the gate with fresh quotes before each purchase.

8 Discussion and Limitations

- **Rental-price dependence.** The first-stage recommendation holds at the $0.25 baseline but not at the $0.20–0.22 marketplace prices (Section 5.5); the relevant price, and the service tier it belongs to, must be chosen deliberately.
- **Time basis and growth accounting.** Costs are 52-week costs (8,736 h) with rates derived from an 8,760-h year. The growth plan is an annuity comparison, not a discounted cash-flow model (Section 5.7).
- **Multi-tenant model assumptions.** Deferrable work is spread uniformly over the whole week rather than scheduled with deadlines and queues, which flatters ownership; purchase timing is partly fixed by gate parameters (no purchase before month 3, projection factor 1.8, size caps, a sale rule); NPV includes the resale value of the fleet at month 36; rents are assumed to fall on average to 85% of

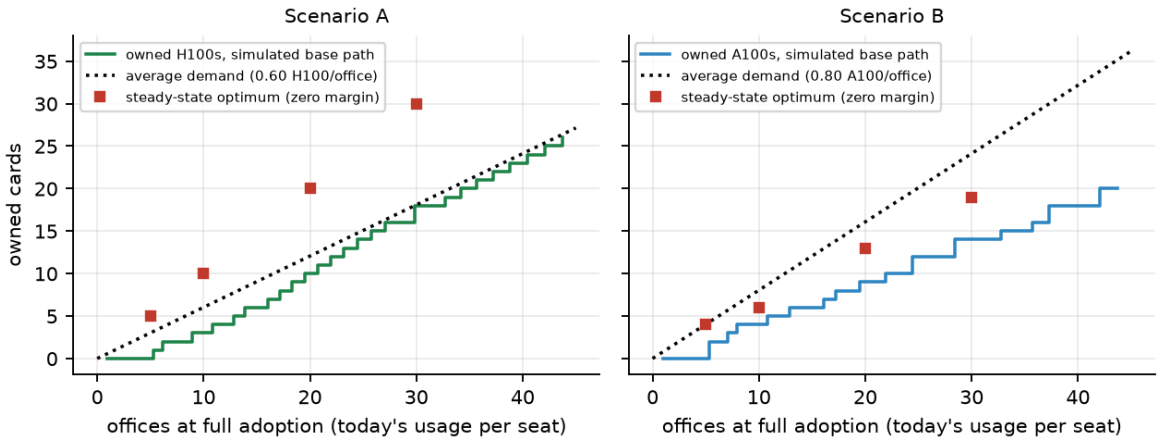

Figure 11: Owned cards along a deterministic onboarding path versus offices at full adoption. Dotted: cards needed for *average* demand; squares: steady-state optimum. The staircase lags the steady-state optimum because purchases wait for evidence of demand.

Table 10: Mean NPV gain, low / high parameter value, k$ (number of losing paths out of 200).

Parameter	Scenario A	Scenario B
Deferrable share 0% / 80%	-2 (116) / 251 (0)	15 (22) / 176 (0)
GPU-s per run 40 / 360 (price scaled)	44 (4) / 448 (0)	36 (7) / 243 (0)
Verifier cost x0.5 / x2 (price scaled)	119 (0) / 249 (0)	83 (0) / 153 (0)
Usage/seat trend -3% / +3% per month	79 (1) / 173 (0)	51 (1) / 142 (0)
New offices/month 0.4 / 1.4	86 (2) / 261 (2)	56 (0) / 162 (0)
Office churn 0.5% / 4% per month	180 (0) / 126 (1)	120 (0) / 86 (1)
Fixed ops cost $250 / $2,500 per month	163 (0) / 94 (19)	109 (0) / 52 (46)
Research/verifier pools merged / separate	166 (0) / 163 (0)	109 (0) / 109 (0)
Trusted-provider rent premium x1.0 / x1.4	163 (0) / 229 (0)	109 (0) / 217 (0)
80GB card prices -30% / +30%	246 (0) / 105 (0)	147 (0) / 68 (5)
80GB rent -30% / +20%	74 (2) / 202 (0)	23 (52) / 163 (0)
Electricity $0.12 / $0.40 per kWh	176 (0) / 153 (0)	130 (0) / 94 (0)
Gate margin 5% / 30%	152 (5) / 176 (0)	109 (0) / 108 (0)

today's level and start above listed marketplace prices (Section 6); the 3090 search stops at 8 cards; the research/verifier split is in whole cards, so a one-card fleet has no verifier capacity, and it is chosen to maximise the large work served rather than to minimise total monthly cost (a service-maximising heuristic; splits that serve equally much but differ in power draw are not distinguished); utilisation-dependent power is applied only to the model's own load-fraction definition (utilisation is measured against available hours, while idle power is charged for all 730 hours); loss counts are reported in full but no confidence intervals are computed (a count of zero in n independent paths bounds the true probability only by about $3/n$ at 95%, i.e. 0.3% for 1,000 paths and 1.5% for 200); and the 12-card share (96%) in the worked examples is computed for the model's own static case only.

- **Throughput equivalences are assumed.** One H100 is taken as 3.5 3090-equivalents in the first stage, 5.5 in scenario A and 3.2 in scenario B of the multi-tenant model (A100: 3.0 in A, 2.4 in B). These are workload-dependent throughput assumptions, not hardware specifications; the preferred 80 GB card (H100 versus A100) is conditional on them until benchmarks with the intended models, quantisation, sequence lengths, batch sizes and inference software are available.

- **Inputs differ between the two stages.** The first-stage model uses electricity at $0.13/kWh, a discount rate of 8%, an H100 at $35,000 per card and $2.50/h rent, and a 3090 rent of $0.25/h before overhead; the multi-tenant cost module uses $0.28/kWh, 12%, an H100 at $26,000 including host and $3.25/h rent, and a 3090 rent of $0.35/h. The two stages should not be compared numerically.

- **Reproducibility.** The first-stage results, figures and manuscript regenerate from `make_paper.py`. The multi-tenant simulator is part of the same script and runs as part of it (unless `--cached` reuses

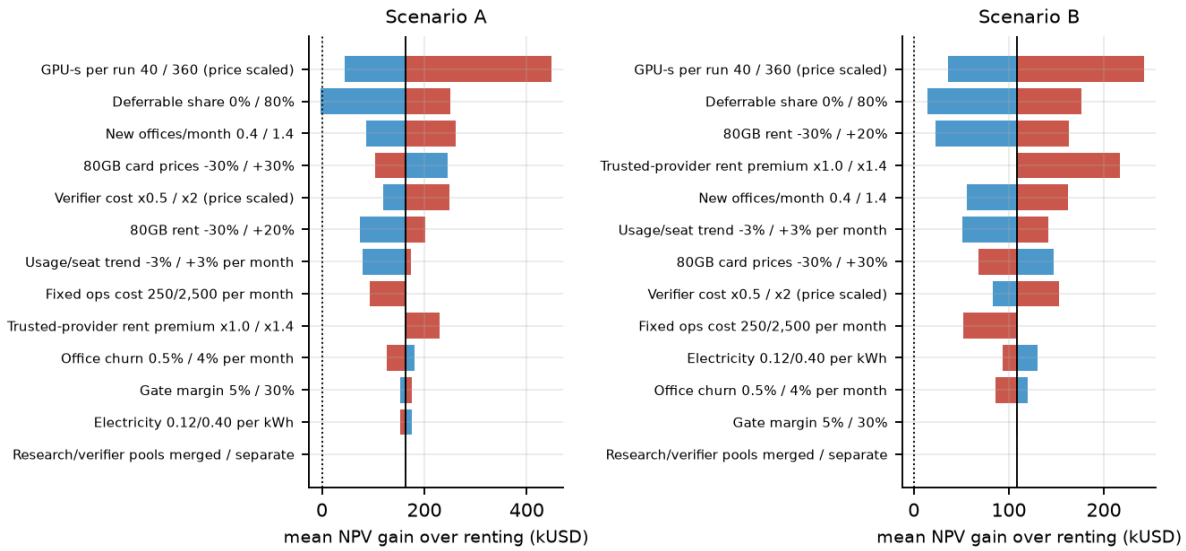

Figure 12: Sensitivity of mean NPV gain (blue = first value, red = second).

Table 11: Example 1: expansion ladder, 3090 class, flat cloud price. Costs in $ per year; "saving" columns are versus all-cloud and versus not expanding.

Year	Load ×	K	Added	Capex added	Cost (optimal)	Cost (all cloud)	Saving vs cloud	Saving vs no exp.
0	1.00	4	2	2,600	8,716	10,677	1,961	184
1	1.35	5	1	1,300	12,133	14,414	2,281	413
2	1.82	7	2	2,600	16,722	19,459	2,736	776
3	2.46	10	3	3,900	22,917	26,269	3,352	1,332
4	3.32	13	3	3,900	31,295	35,463	4,168	2,103
5	4.48	18	5	6,500	42,588	47,875	5,287	3,187

a previous log), so these results are in principle reproducible. The multi-tenant results reported here come from a full-size run (1,000 Monte Carlo paths per scenario with seed 7, 200 per sensitivity case with seed 11; script hash dc297222ce3e123a in the log header) of the corrected allocation and power accounting, giving mean NPV gains of $159k (A, load-aware H100 gate) and $104k (B, load-aware A100 gate); they still rest on the unvalidated inputs listed here, so treat them as scenario results, not forecasts.

- **Corrections relative to the previous version.** (i) Maintenance on existing cards had been removed together with their sunk capital charge; restoring it adds $129.64 per 52 weeks for the two existing cards and changes the headline savings from 19.6%, 2.1% and 60.3% to 18.4%, 2.1% and 59.9% (versus all-cloud, versus keeping K_0, versus 99.9th-percentile sizing); the optimal K is unchanged. (ii) The growth table had treated every earlier purchase as sunk; it now charges capital recovery and maintenance on all cards beyond K_0. (iii) The integer-capacity statement of the fractile rule was only exact for integer-valued load; it is replaced by Proposition 2. (iv) The analytic rule returned unbounded capacity when the cloud price did not exceed dynamic energy cost; it now returns no new capacity. (v) The build script failed when the multi-tenant log was missing; it now parses the log, falls back to cached results, or stops with a clear message. (vi) The user roadmap was described as buying 4 H100s and 8 3090s from month 0; in the cost module it buys two 3090s in month 0 and the rest through utilisation triggers. (vii) Example card costs now come from the cost module rather than from a capex-per-fleet ratio (the values coincide). (viii) The statements that the multi-tenant simulator script was unavailable were removed: the simulator is part of the merged script, and only runs with `--cached` reuse an earlier log. (ix) The multi-tenant allocation now assigns 80 GB cards to the research and verifier pools in whole cards by the split that serves most work (previously $\lfloor n/2 \rfloor$ cards, with capacity taken pro rata to card count) and computes power per card type and pool (previously one pooled utilisation for every 80 GB card); the model was re-run at full size (headline mean gains $159k in A and $104k in B). (x) In this revision: the abstract now says "99.9th-percentile load"

Table 12: Example 2: expansion cases for 20 offices (k$). Costs are monthly and include the capital charge; margin = revenue − cost to serve (% of revenue); last column is the annual cost saving. The 12-card row is priced directly in the cost model.

Case	Cards	Capex	Revenue/mo	Rent-only cost/mo	Own+burst cost/mo	Margin rent-only	Margin owning	Saving/yr
A: 12 H100 (0.6 per office)	12	312.0	50.0	29.1	19.3	20.9 (42%)	30.7 (61%)	117.2
A: 20 H100 (full optimum)	20	520.0	50.0	29.1	18.9	20.9 (42%)	31.1 (62%)	122.4
B: 13 A100 (full optimum)	13	208.0	50.0	22.1	16.2	27.9 (56%)	33.8 (68%)	70.8

Table 13: Example 3: staged expansion, scenario A, 0.6 H100 per office, 40% deferrable (k$).

Offices	Revenue/mo	H100 cards	Capex	Net saving/mo	Saving / revenue
5	12.5	3	78.0	2.4	20%
10	25.0	6	156.0	4.9	20%
20	50.0	12	312.0	9.8	20%
30	75.0	18	468.0	14.7	20%

rather than "peak" for the sizing comparison; the scenario B headline is the A100 gate's $104k (an earlier $103k belonged to the all-types gate); loss probabilities are reported as raw counts instead of rounded percentages; the contradictory statements about the full-size run and the seeds were replaced by the provenance line above; the 12-card share is computed rather than assumed; the pool split is described as a service-maximising heuristic; Proposition 2 states $p_c > e$; and a garbled sentence in Section 2 and the DGX per-GPU power wording were corrected.

- **Assumed inputs.** Workload and prices are synthetic/assumed (Section 4); the multi-tenant extension is parsed from the log of an earlier run, with the inputs listed in Table 6, so its absolute dollar figures are only as good as those inputs. Conclusions about specific dollar savings are illustrative; the *method* and the qualitative sensitivities are the contribution.
- **Fungibility.** Load is treated as perfectly divisible across owned and rented GPUs. Multi-GPU sessions, VRAM limits, interconnect and data-transfer latency make some overflow infeasible.
- **No queuing.** Instead of renting at peak, an office may delay jobs; adding a delay-cost term turns p_c into $\min(p_c, \text{delay cost})$ and lowers the overflow price, shifting K^* downward.
- **Tenant-side effects.** Service-level agreements, tenant confidentiality requirements and correlated demand shocks (e.g. a model release that changes GPU-seconds per run) are only partly captured.
- **Resale and reliability.** In the first-stage model, resale value, failures, spot interruptions and reserved-instance discounts are not modelled (the multi-tenant simulator does value the fleet at resale: cards sell at a 10% discount, 80 GB cards lose 15% of value per year and their resale scales with the rent index, so its NPV gains include terminal resale value, and its gate sells cards whose utilisation is below 30% after month 12); reserved cloud prices act like an additional capacity tier with its own c_f, and the screening-curve logic extends to it.
- **Power and space.** Facility power and cooling bound owned capacity regardless of financial optimality; DGX-class nodes need dedicated provisioning [13, 14]. The model assumes the provider can host the fleet it buys.

9 Conclusion

For bursty small-office agent workloads, ownership should cover the base of the load-duration curve down to the point where the probability of exceeding capacity equals $c_f/(p_c - e)$, and everything above should be rented. With the stated assumptions this means owning 4 units, a decision that is robust to workload noise but sensitive to the cloud price, which should be monitored continuously: at the lower marketplace prices listed in November 2026 the expansion is not justified. For a data center serving small and medium offices, the multi-tenant extension adds three operational conclusions: buy against the load-duration curve rather than average demand, wait until several tenants are on board (the simulated gate enforces a minimum waiting period, so this is a design choice and not a tested optimum), and actively sell customers on deferrable

(queued) work, which is the strongest lever on the value of owned hardware in a model that smooths such work uniformly across the week. These multi-tenant conclusions hold in the full-size run of the corrected simulator, but remain hypotheses until tested with realistic rental prices, benchmarked throughput and stricter deferral constraints.

References

[1] Vast.ai, "How much does it cost to rent a GPU in the cloud," https://vast.ai/blog/how-much-does-it-cost-to-rent-a-gpu-in-the-cloud.

[2] Vast.ai, "Live GPU pricing guide," https://vast.ai/article/how-much-does-it-cost-to-rent-a-gpu-in-the-cloud-live-pricing-guide.

[3] Vast.ai, "Historical pricing," https://vast.ai/oldpricing.

[4] GPUPerHour, "RTX 3090 rental," https://www.gpuperhour.com/rent/rtx-3090.

[5] GetDeploying, "NVIDIA RTX 3090," https://getdeploying.com/reference/cloud-gpu/nvidia-rtx-3090.

[6] Flopper, "RTX 3090 rental pricing," https://flopper.io/pricing/nvidia-geforce-rtx-3090.

[7] DeployBase, "RTX 3090 on Vast.ai," https://deploybase.ai/articles/rtx-3090-vastai.

[8] RunPod, "RTX 3090," https://www.runpod.io/gpu-models/rtx-3090; and "AI server cost," https://runpod.io/articles/guides/ai-server-cost.

[9] DEV Community, "Cloud GPU pricing compared: RunPod vs Vast.ai vs Lambda Labs," https://dev.to/jovan_chan_9500711396d4e6/cloud-gpu-pricing-compared-runpod-vs-vastai-vs-lambda-labs-2026-35bn.

[10] Tech Insider, "RunPod vs Lambda vs Vast.ai," https://tech-insider.org/runpod-vs-lambda-vs-vast-ai-2026/.

[11] Neurohive, "Cloud GPU pricing in 2026," https://neurohive.io/en/guides/cloud-gpu-pricing/.

[12] NVIDIA, "DGX H100," https://nvidia.com/en-us/data-center/dgx-h100; Dell, "NVIDIA DGX H100 datasheet," https://i.dell.com/sites/csdocuments/App-Merchandizing_Documents/en/us/nvidia-dgx-h100-datasheet.pdf.

[13] NVIDIA, "DGX SuperPOD electrical specifications," https://docs.nvidia.com/dgx-superpod/design-guides/dgx-superpod-data-center-design-h100/latest/electrical.html.

[14] NVIDIA, "Data center deployment planning," https://docs.nvidia.com/dgx-superpod/design-guides/dgx-superpod-data-center-design-h100/latest/planning.html.

[15] NVIDIA, "DGX H100/H200 system user guide," https://docs.nvidia.com/dgx/dgxh100-user-guide/.

[16] Markaicode, "Vast.ai pricing (November 2026)," https://markaicode.com/pricing/vast-ai-pricing/ (live prices read 16 November 2026).

[17] ComputePrices, "RTX 3090 pricing on Runpod," https://computeprices.com/providers/runpod/gpus/rtx3090.

[18] arXiv:2407.14645, https://arxiv.org/pdf/2407.14645. (Listed in the supplied reading list; not used quantitatively.)