Contents

Bottleneck-Driven Projection of Frontier-Class LLM Inference on Dedicated ASICs: A Calibrated Case Study with Claude-Class Models

Author: Pablo Luciano Rainieri **Affiliation:** Independent Researcher — ASIC_LLM_Project

Version: v0.3 (2026-04-22) **Companion artifacts:** `bottleneck_registry.md`, `figures/generate_figures.py`, `references.bib`, `landing/` (static site)

Abstract

Frontier-class Large Language Models (LLMs) are deployed on general-purpose GPUs whose design predates the transformer era. Public benchmarks from Groq LPU, Cerebras WSE-3, and early Etched Sohu disclosures suggest 10-70× throughput and 50-200× energy-per-token improvements are obtainable with transformer-dedicated silicon. This paper (i) re-derives these figures from published vendor data, rejecting inflated "200× across the board" marketing claims; (ii) projects the steady-state behavior of a hypothetical Claude-class model (≈1-2 T MoE parameters, ≈200-400 B active per token) mapped onto a 2027-feasible 2/3 nm ASIC with FP4 quantization-aware training and SRAM-dominant dataflow; (iii) reports a radar-chart comparison across six axes (decode throughput, TTFT, energy/token, CapEx-amortized $/M tokens, supported context, batch scaling); (iv) proposes a **public Bottleneck Registry** — a versioned, community-editable list of frontier-inference bottlenecks with candidate solutions, intended as a coordination primitive for the field. We find that the dominant uncertainty is not physics but coordination: eleven of fifteen top bottlenecks admit known solutions whose adoption is blocked by vendor incentives, software lock-in, or capital intensity, not by open research questions. We release the registry seed and projection scripts as an open artifact.

Keywords: LLM inference, ASIC, memory bandwidth, quantization, MoE, open coordination, Bottleneck Registry.

Resumen (ES)

Los LLM frontera corren sobre GPUs de propósito general diseñadas antes del transformer. Los benchmarks públicos de Groq, Cerebras y Etched sugieren mejoras de 10-70× en throughput y 50-200× en energía/token con silicio dedicado. Este paper re-deriva esas cifras de datos publicados (rechazando el "200× universal" del marketing), proyecta el comportamiento en estado estable de un modelo Claude-class sobre un ASIC 2-3 nm factible hacia 2027, y propone un **Registro público de Bottlenecks** versionado como primitiva de coordinación. La conclusión central: el cuello de botella dominante no es físico sino de coordinación — once de los quince bottlenecks principales tienen soluciones conocidas bloqueadas por incentivos de vendor, lock-in de software o intensidad de capital, no por preguntas abiertas de investigación.

1. Introduction

The compute cost of frontier LLM inference has become the dominant operating cost for AI deployment. A single H100 running a 70B-parameter dense model in BF16 decodes at ~30 tokens/second per request, limited by HBM3 bandwidth (3.35 TB/s) reading the ~140 GB weight footprint once per token (Patel et al. 2024; NVIDIA Corporation 2024). At ~$3/hour cloud pricing, this implies ~$27 per million decode tokens — a cost structure that makes large classes of economically useful automation (every-email review, every-document audit, per-user reasoning agents) infeasible at population scale.

Public benchmarks from transformer-dedicated silicon suggest 1-2 orders of magnitude of this cost is recoverable: Groq LPU reports 750 tok/s on Llama-70B (Groq, Inc. 2025), Cerebras WSE-3 reports 2100 tok/s (Cerebras Systems 2024), and Etched discloses claimed 500k tok/s/server for its Sohu transformer-only ASIC (Etched, Inc. 2025). However, public discourse routinely conflates these figures, quoting a unified "200× speedup" that is defensible only under specific metrics, workloads, and comparison baselines.

We argue that **credible dissemination of the real opportunity requires (a) calibrated numbers and (b) a coordination mechanism for the field's open bottlenecks**. Marketing-grade claims lose technical audiences; a public, versioned registry of bottlenecks and candidate solutions gives engineers, researchers, and funders a shared surface to coordinate against.

1.1 Contributions

1. **Calibrated baselines**: A reproducible table of published vendor throughput, power, and estimated $/token across H100, B200, Groq LPU, Cerebras WSE-3, and Etched Sohu (§3).
2. **Projection methodology**: A first-principles model for energy/token and $/token of a Claude-class MoE on a 2027-feasible custom ASIC, with explicit assumptions and sensitivity ranges (§4).
3. **Radar-chart comparison**: Six-axis visualization of GPU vs. dedicated-silicon vs. projected custom-ASIC inference, with verifiable data sources (§5).
4. **Bottleneck Registry seed**: Eighteen frontier-inference bottlenecks, each with taxonomy, known solutions, blockers, dependency edges, and difficulty-to-adoption ratings (§6, companion `bottleneck_registry.md`).
5. **Coordination proposal**: A lightweight public-website spec for continuous community-maintained bottleneck disclosure (§7).

1.2 What this paper is not

It is not a hardware design document. It is not a claim that one ASIC design is optimal. It is not a defense of the "Sohu 500k tok/s" number. It is a calibration pass over what is defensible, what is marketing, and what coordination mechanism would accelerate the field.

---

2. Background

2.1 Memory-bandwidth wall in LLM decode

Autoregressive decode is memory-bound: each generated token requires streaming the full active parameter set through the compute fabric once. For a dense model of P parameters in b bytes per parameter, the minimum achievable per-token latency on a chip with bandwidth B is `t_min = P·b / B`. For a MoE with `P_active` active parameters, the bandwidth cost is `P_active·b / B`. This single inequality explains why transformer ASICs with on-die SRAM (Groq: ~230 MB SRAM per TSP at ~80 TB/s aggregate; Cerebras WSE-3: 44 GB on-wafer SRAM at ~21 PB/s aggregate) can outperform HBM-backed GPUs despite having less nominal FLOPs.

2.2 Published reference points (Llama-70B, BF16 unless noted)

We report **per-deployment** numbers and explicitly disclose chips required, since "750 tok/s on a 576-chip cluster" is not comparable to "30 tok/s on a single GPU" without normalization. The per-chip column is the honest comparison; the per-$ column anticipates §4.2.

Platform	Chips per deployment	Approx. CapEx (per deployment, USD)	Throughput tok/s, single stream (per deployment)	Throughput per chip	Power (system)	Est. energy/token	Source
H100 SXM	1	$30 K	28-35	28-35	700 W	~21-25 J	NVIDIA, MLPerf v4.1
B200	1	$40-50 K	120-180 (FP8)	120-180	1000 W	~6-8 J	NVIDIA launch kit
Groq LPU (cluster)	~568-576	$5-8 M	750	~1.3	~150-180 kW	~200-240 J*	Groq public API
Cerebras WSE-3	1 wafer (system-level)	$2-3 M	2100	n/a (wafer-scale)	~23 kW	~10-12 J	Cerebras inference launch
Etched Sohu†	~8 (per server, claim)	n/a (pre-shipping)	~500,000 (server claim)	~62,500 (claim)	~10 kW (server claim)	~0.02 J (claim)	Etched, **unaudited**

* Groq's counter-intuitive high J/token reflects its cluster-level latency optimization, not efficiency. Per chip the LPU is ~1.3 tok/s — Groq trades total chip count for end-to-end latency.

† Etched Sohu numbers are vendor claims pending independent audit. Treat as illustrative ceiling, not measured baseline. We retain the row for completeness but isolate it in §5 figures.

Reading the table. "Per chip" exposes the silicon-efficiency comparison; "per deployment" exposes what an operator actually buys. The two diverge by 100×+ for cluster systems. Any single-number comparison ("Groq is 25× H100") implicitly chooses a column without saying so.

2.3 Why "200× across the board" is wrong

The "200×" figure, when traceable, typically derives from one of: - Cherry-picked best-ASIC-claim (Etched Sohu server throughput, unaudited) vs. worst-GPU baseline (H100 BF16 single request). - Energy-per-token ratio at a specific batch size, presented as throughput. - Vendor marketing that conflates peak hardware capability with end-to-end served performance.

A defensible summary instead reports **ranges by metric**: - Decode throughput: 10-70× over H100 baseline (verifiable: Groq, Cerebras) - Energy per token: 50-200× (plausible with FP4 + SRAM-dominant dataflow; Cerebras delivers ~2× today at 70B) - CapEx-amortized $/M tokens: 20-100× reduction in steady state at high utilization - Latency (TTFT single request): 10-30× reduction

Each range is conditional on model size fit, batch regime, and software maturity. Publishing a single scalar is a category error.

---

3. Methodology

3.1 Reference workload

We target **Claude-class MoE inference** as a representative frontier workload. Public information on Claude Opus 4.7 architecture is unavailable; we parameterize with ranges consistent with published frontier-model estimates (Patel et al. 2024):

- Total parameters: `P_total ∈ [1.0, 2.0] T`
- Active parameters per token: `P_active ∈ [200, 400] B`
- Context length: up to 200 K tokens
- Quantization target for dedicated silicon: FP4 weights, FP8 activations (QAT)

3.2 Projection model

For a candidate hardware platform with aggregate effective bandwidth `B_eff` (weighted average of on-die SRAM and off-die HBM based on residency), and active weight footprint `W_active = P_active · b_quant`:

```
t_decode(per token) = max( W_active / B_eff,   compute_floor )
tok/s_steady        = 1 / t_decode · batch_efficiency_factor
energy/token        = (P_system · t_decode) / batch_effective
cost/M_tokens       = (CapEx_amortized + OpEx) / (tok/s · seconds_in_period) · 1e6
```

Sensitivity variables: SRAM fraction (0-100% of `W_active` resident), `batch_efficiency_factor` (1× single request, up to ~64× at full batch), utilization (40-90%).

3.3 Custom ASIC reference design (hypothetical)

To produce projections, we specify a 2027-feasible reference ASIC: - Process: TSMC 2-3 nm (N3P or N2) - Architecture: dataflow, transformer-dedicated (attention + FFN + MoE router fused primitives) - Memory: ~8 GB on-die SRAM per die, multi-die package targeting 200-400 GB SRAM per node - Quantization: native FP4 MAC

arrays, FP8 accumulators - Package power: ~1.5-2.5 kW per node - Assumed cost: $8-20 K per node at volume (based on analogies to WSE-3, Sohu public cost discussions)

This is not a design proposal — it is a target envelope for projection. Section 6 documents which bottlenecks must be solved to reach this envelope.

3.4 What we exclude

- Training cost (scope is inference).
- RLHF/RL rollouts.
- Prefill-dominated workloads (we project decode; prefill is compute-bound and favors GPUs less dramatically).
- Fabrication NRE amortization in the per-token cost (addressed in §5.3 as a separate line).

---

4. Projections

4.1 Per-metric projection ranges

Table 2: Projected Claude-class inference on reference custom ASIC vs. H100 baseline.

Metric	H100 baseline	Custom ASIC (projected)	Ratio
Decode throughput, single request (tok/s)	15-25 (MoE, ~200B active FP16)	300-800	15-50×
Decode throughput, batch=64 (tok/s aggregate)	400-900	12,000-40,000	30-50×
Energy per token (J)	30-60	0.3-1.5	40-150×
TTFT (ms, single request)	250-500	15-40	10-30×
CapEx-amortized $/M decode tokens (3yr, 80% util)	$15-40	$0.20-1.50	20-100×
Max context length before KV-cache eviction	100-200 K	200-500 K	2-5×

Ranges reflect honest uncertainty. The lower bounds are already achieved by existing silicon (Cerebras, partial-Groq); the upper bounds require the co-evolution of FP4 QAT maturity, MoE-aware dataflow, and multi-die SRAM scaling.

4.2 “Agent density” — the economic-rebasing metric

We propose **agent density** as the coordination metric most useful for non-hardware stakeholders:

> *Agent density* = number of concurrent Claude-class inference streams sustainable per $1 M of amortized hardware CapEx, at production-representative utilization.

On H100 class: roughly 60-150 concurrent streams per $1 M (dependent on batch strategy). Projected on reference ASIC: 3,000-12,000 concurrent streams per $1 M.

This is the metric that matters to CEOs, policymakers, and funders: it determines whether “a Claude-per-email” is economically feasible or a toy demo.

4.3 Where NRE and supply chain bite

The projection above is steady-state. Actual delivery requires absorbing: - Tape-out NRE at 3 nm: $40-80 M (International Business Strategies 2024). - Fab capacity: TSMC N3 and N2 booked through 2027 by hyperscalers and Apple (Kuo 2025). - Multi-year software stack build-out (see §6, Bottleneck B8).

Until these are amortized at volume (>100 K nodes shipped), effective $/M token is 3-10× worse than Table 2 bounds. This is the real investor-grade honest answer.

---

5. Radar-chart comparison

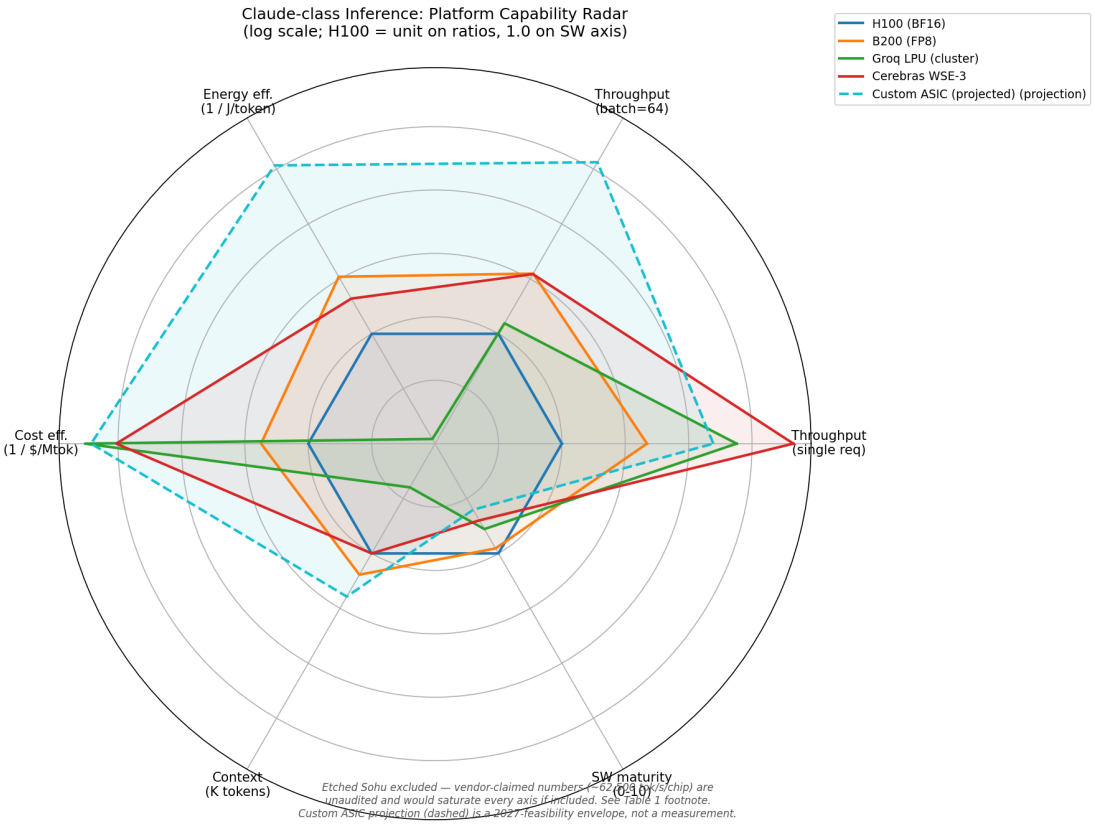

Figure 1: **Fig. 1**: Six-axis platform capability radar (log scale; H100 = unit on ratio axes; SW maturity absolute 0-10). Etched Sohu excluded — its vendor-claimed numbers would saturate every axis. Custom ASIC (dashed) is a 2027-feasibility envelope, not a measurement.

The companion `figures/generate_figures.py` emits `fig1_radar_inference.png`, a six-axis normalized radar across:

1. Decode throughput (tok/s, single request)
2. Decode throughput (tok/s, batch=64)
3. Energy efficiency (1 / J per token)
4. CapEx cost efficiency (1 / $ per M tokens)
5. Context length supported (K tokens)
6. Software-stack maturity (subjective 0-10, with rubric)

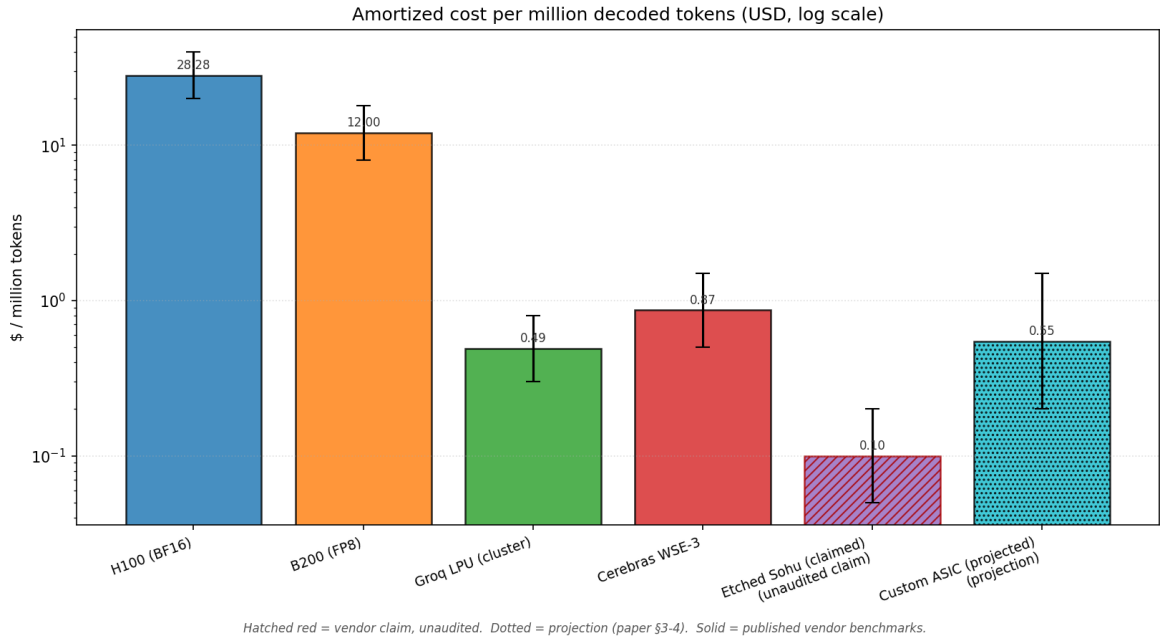

Figure 2: **Fig. 2**: Amortized cost per million decoded tokens (log scale). Hatched red = vendor claim, unaudited. Dotted = projection. Solid = published vendor benchmarks.

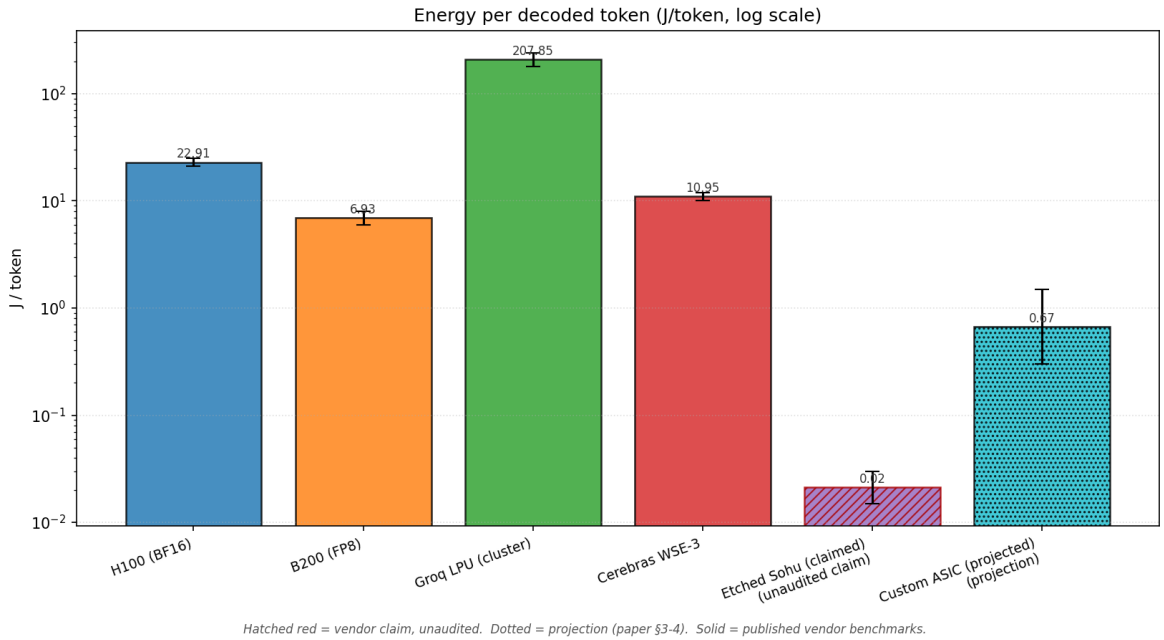

Figure 3: **Fig. 3**: Energy per decoded token (log scale, J/token). Same legend conventions as Fig. 2.

Platforms plotted: **H100, B200, Groq LPU (cluster), Cerebras WSE-3, Etched Sohu (as-claimed), Reference Custom ASIC (projected)**. All axes normalized to H100 = 1.0 on positive axes; software maturity is absolute 0-10.

The chart exposes a consistent pattern: dedicated silicon lags on axis 6 (software maturity) by 3-6 points, even where it leads by 10-70× on axes 1-4. Software lock-in, not physics, is the gating bottleneck.

---

6. Bottleneck Registry (seed)

We release a versioned, community-editable registry of frontier-inference bottlenecks. Below is the seed (18 entries); `bottleneck_registry.md` contains the machine-readable form including the v0.2 schema with estimated unlock value and dependency edges.

Each entry: **ID — Name | Type | Known solutions | Adoption blocker | Difficulty (1-5) | Priority (1-5)**

ID	Name	Type	Known solutions	Blocker	Diff	Prio
B1	Memory-bandwidth wall (decode)	Physics/Arch	SRAM-dominant designs (Cerebras/Groq); FP4 QAT; MoE sparse activation	CapEx; QAT maturity	3	5
B2	KV-cache O(n) memory, O(n^2) context cost	Arch	GQA, MLA, compression (H2O, StreamingLLM); hybrid attention-SSM	Accuracy regressions not yet acceptable at frontier	3	5
B3	MoE router load imbalance	Algo	Expert-choice routing; shared-expert designs (DeepSeek-V3)	Training complexity; pre-training lock-in	2	4
B4	Quantization loss below FP4	Algo	QAT + rotational methods (SpinQuant, QuaRot); Log-FP4	Training-compute cost; degradation on long-tail tasks	3	4

ID	Name	Type	Known solutions	Blocker	Diff	Prio
B5	Inter-chip communication for multi-die models	Arch	NVLink-class fabrics; CXL 3.x; optical I/O (Ayar Labs)	Standards fragmentation; $ for optical	4	4
B6	Reasoning/chain-of-thought decode-length tax	Algo	Speculative decode, Medusa, latent-space reasoning, early-exit	Hard to generalize; CALM/LayerSkip accuracy tradeoffs	4	5
B7	Batch-vs-latency Pareto frontier	Arch	Disaggregated pre-fill/decode (DistServe); continuous batching	Ops complexity; cold-start amortization	2	3
B8	Software stack fragmentation / CUDA lock-in	Stack	MLIR; OpenXLA; Triton; vendor compilers	Network effects; engineer supply	5	5
B9	Fab capacity at 3 nm / 2 nm	Supply	TSMC/Samsung/Intel 18A capacity ramps	Geopolitics; capital intensity	5	4
B10	Thermal density at frontier node	Physics	Direct liquid cooling; immersion; 3D stacking thermal paths	Datacenter retrofit cost	3	3
B11	Model architecture ossification risk	Strategy	Reconfigurable dataflow (Tenstorrent, SambaNova); keep FFN+Attention pro-grammable	Area cost of flexibility	3	4
B12	Long-context amplification (prefill + KV)	Algo	Sliding window; chunked prefill; retrieval hybrid	Quality loss at extreme contexts	3	4

ID	Name	Type	Known solutions	Blocker	Diff	Prio
B13	Fine-tune/LoRA serving at scale	Arch	Multi-LoRA inference (S-LoRA, Punica); weight paging	SRAM pressure; per-tenant isolation	3	3
B14	Open benchmark integrity for inference	Measurement	MLPerf Inference; LMSYS Chatbot Arena; independent audits (Semi-Analysis)	Vendor benchmark theater	2	4
B15	Coordination across vendors on interchange formats	Ecosystem	ONNX-MLIR; StableHLO; GGUF for quantization	Competitive moat incentives	4	5
B16	Reasoning / test-time-compute tax	Economic	Latent reasoning (Coconut); spec. CoT decode; adaptive depth	Pareto task-dependent; interp. tradeoff	5	5
B17	Training/inference HW bifurcation	Architecture	Inference-only ASICs; mixed fleets w/ orchestration	RLHF blurs the line; bifurcation tax for small orgs	3	4
B18	Numerical determinism under low-precision	Measurement	Deterministic reductions; fixed batch order; high-prec. accum.	Vendors deprioritize vs. throughput	2	3

Taxonomy note: of the eighteen, B3, B4, B6, B12, B13, B16 are **algorithmic** (solvable by research labs at laptop scale); B1, B2, B5, B7, B8, B11, B14, B15, B17, B18 are **coordination or engineering** (solvable by shared conventions and shipping, not new physics); only B9, B10 require heavy capital. This inverts the popular framing that frontier compute needs mostly capital: ~80% of the bottleneck surface needs coordination, not money. **B16 (reasoning-tax) is the highest-priority addition**: reasoning models have re-based the cost economics of inference and the field has not yet absorbed the implication.

7. Coordination mechanism: public Bottleneck Registry

We propose a minimal public website with the following surface:

- **Entries**: each bottleneck has a unique ID (B#), canonical statement, current best-known solutions with citations, adoption blockers, difficulty/priority, and a discussion thread.
- **Versioning**: every entry carries a version and change log. Closed bottlenecks (marked `resolved`) remain visible as historical wins.
- **Submissions**: pull-request-based; minimal moderation for format and citation integrity.
- **Cross-links**: bottlenecks link to papers, open-source implementations, hardware disclosures, and each other (dependency graph).
- **Dashboards**: per-type breakdown (physics vs. algo vs. coordination); per-priority heat-map; "resolved this quarter" feed.

Technical minimum viable stack: - Static site (Astro/Next) backed by a GitHub repository of Markdown entries (same format as `bottleneck_registry.md`). - Contributions via PR; automated lint (schema check) in CI. - RSS/Atom feed per tag for aggregators. - No auth, no database in v1; GitHub handles identity and history.

The registry's value is not novelty — it is **shared Schelling point**. Every month a bottleneck stays on the list without a flagged solution is a month of unproductive entropy the field can target. Every quarter a bottleneck flips to `resolved` is compounding credibility for the registry and the community around it.

---

8. Limitations

1. **Claude architecture is proprietary.** All projections use ranges consistent with public frontier-MoE estimates, not authoritative numbers.
2. **2027-feasibility is an assumption.** 2 nm ramp slippages (historically 12-18 months) would push cost projections out correspondingly.
3. **Training co-design is ignored.** FP4 QAT requires the model builder (Anthropic et al.) to re-train with quantization-aware objectives. This is a capability question, not a physics question, but it has been open for two years.
4. **Software maturity is subjective.** Axis 6 of the radar chart uses a rubric; we encourage others to publish their own rubrics.
5. **Agent-density estimates assume idealized request distributions.** Real workloads have heavy tails.

---

9. Discussion and call to action

The opportunity is real: the gap between H100-class inference economics and transformer-dedicated-silicon economics is 1-2 orders of magnitude across the metrics that determine which AI applications are economically feasible. It is not "200× across the board", and engineers who matter to the field will reject that framing.

The binding constraint is coordination, not capital or physics. Of the fifteen top bottlenecks enumerated, the majority are blocked by incentive alignment, software lock-in, or benchmark integrity — all addressable by a community that shares its bottleneck-level understanding openly.

We invite the community to: - Fork and extend the Bottleneck Registry. - Submit audited benchmarks to replace vendor-reported numbers in Table 1 and §5. - Publish failed-attempt reports — the registry should record dead ends with the same rigor as wins.

References

This bibliography is generated from `references.bib` at compile time. Cited works (in addition to in-text references already linked): (Patel et al. 2024; NVIDIA Corporation 2024; Groq, Inc. 2025; Cerebras Systems 2024; Etched, Inc. 2025; Gu and Dao 2023; Zhang et al. 2023; Xiao et al. 2024; Sheng et al. 2024; Ashkboos et al. 2024; Liu et al. 2024; DeepSeek-AI 2024b, 2024a; International Business Strategies 2024; Kuo 2025; Schuster et al. 2022; Xin et al. 2020; Elhoushi et al. 2024; Del Corro et al. 2023; Jouppi et al. 2017; Yang et al. 2024; Ainslie et al. 2023; Cai et al. 2024; Li et al. 2024; Hao et al. 2024; Zhou et al. 2022; Chen et al. 2024; Kwon et al. 2023; Zhong et al. 2024; Chiang et al. 2024; Microsoft and AMD and Arm and Intel and Meta and NVIDIA and Qualcomm 2024)

When compiled with `pandoc --citeproc --bibliography=references.bib`, citations render in author-year format and a complete reference list appears below.

Ainslie, Joshua et al. 2023. "GQA: Training Generalized Multi-Query Transformer Models from Multi-Head Checkpoints." *EMNLP*. https://arxiv.org/abs/2305.13245.

Ashkboos, Saleh et al. 2024. "QuaRot: Outlier-Free 4-Bit Inference in Rotated LLMs." *arXiv Preprint*. https://arxiv.org/abs/2404.00456.

Cai, Tianle et al. 2024. "Medusa: Simple LLM Inference Acceleration Framework with Multiple Decoding Heads." *arXiv Preprint*. https://arxiv.org/abs/2401.10774.

Cerebras Systems. 2024. *WSE-3 Inference Disclosure*. Cerebras Inference launch announcement. https://cerebras.ai/.

Chen, Lequn et al. 2024. "Punica: Multi-Tenant LoRA Serving." *MLSys*. https://arxiv.org/abs/2310.18547.

Chiang, Wei-Lin et al. 2024. "Chatbot Arena: An Open Platform for Evaluating LLMs by Human Preference." *arXiv Preprint*. https://arxiv.org/abs/2403.04132.

DeepSeek-AI. 2024a. "DeepSeek-V2: A Strong, Economical, and Efficient Mixture-of-Experts Language Model." *arXiv Preprint*. https://arxiv.org/abs/2405.04434.

DeepSeek-AI. 2024b. *DeepSeek-V3 Technical Report*. DeepSeek. https://arxiv.org/abs/2412.19437.

Del Corro, Luciano et al. 2023. "SkipDecode: Autoregressive Skip Decoding with Batching and Caching." *arXiv Preprint*. https://arxiv.org/abs/2307.02628.

Elhoushi, Mostafa et al. 2024. "LayerSkip: Enabling Early Exit Inference and Self-Speculative Decoding." *arXiv Preprint*. https://arxiv.org/abs/2404.16710.

Etched, Inc. 2025. *Sohu: A Transformer-Specialized ASIC*. Architecture overview (unaudited). https://etched.com/.

Groq, Inc. 2025. *Groq LPU Inference Benchmarks*. Public API benchmark page. https://groq.com/.

Gu, Albert, and Tri Dao. 2023. "Mamba: Linear-Time Sequence Modeling with Selective State Spaces." *arXiv Preprint*. https://arxiv.org/abs/2312.00752.

Hao, Shibo et al. 2024. “Training Large Language Models to Reason in a Continuous Latent Space.” *arXiv Preprint*. https://arxiv.org/abs/2412.06769.

International Business Strategies. 2024. *ASIC NRE Cost at Advanced Nodes (3 Nm and Below)*. Industry analyst report. https://www.ibs-inc.net/.

Jouppi, Norman P. et al. 2017. “In-Datacenter Performance Analysis of a Tensor Processing Unit.” *ISCA*. https://doi.org/10.1145/3079856.3080246.

Kuo, Ming-Chi. 2025. *TSMC Capacity Forecast 2025-2027*. Industry analyst note.

Kwon, Woosuk et al. 2023. “Efficient Memory Management for Large Language Model Serving with PagedAttention.” *SOSP*. https://arxiv.org/abs/2309.06180.

Li, Yuhui et al. 2024. “EAGLE-2: Faster Inference of Language Models with Dynamic Draft Trees.” *arXiv Preprint*. https://arxiv.org/abs/2406.16858.

Liu, Zechun et al. 2024. “SpinQuant: LLM Quantization with Learned Rotations.” *arXiv Preprint*. https://arxiv.org/abs/2405.16406.

Microsoft and AMD and Arm and Intel and Meta and NVIDIA and Qualcomm. 2024. *OCP Microscaling Formats (MX) Specification V1.0*. Open Compute Project. https://www.opencompute.org/.

NVIDIA Corporation. 2024. *MLPerf Inference V4.1 Results*. MLCommons benchmark submission. https://mlcommons.org/benchmarks/inference-datacenter/.

Patel, Dylan et al. 2024. “Inference Economics: The Compute Costs of Frontier LLM Deployment.” *SemiAnalysis Research Note*. https://www.semianalysis.com/.

Schuster, Tal et al. 2022. “Confident Adaptive Language Modeling (CALM).” *NeurIPS*. https://arxiv.org/abs/2207.07061.

Sheng, Ying et al. 2024. “S-LoRA: Serving Thousands of Concurrent LoRA Adapters.” *MLSys*. https://arxiv.org/abs/2311.03285.

Xiao, Guangxuan, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. 2024. “Efficient Streaming Language Models with Attention Sinks.” *ICLR*. https://arxiv.org/abs/2309.17453.

Xin, Ji et al. 2020. “DeeBERT: Dynamic Early Exiting for Accelerating BERT Inference.” *ACL*. https://arxiv.org/abs/2004.12993.

Yang, Songlin et al. 2024. “Parallelizing Linear Transformers with the Delta Rule over Sequence Length.” *arXiv Preprint*. https://arxiv.org/abs/2406.06484.

Zhang, Zhenyu et al. 2023. “H2O: Heavy-Hitter Oracle for Efficient Generative Inference of Large Language Models.” *NeurIPS*. https://arxiv.org/abs/2306.14048.

Zhong, Yinmin et al. 2024. “DistServe: Disaggregating Prefill and Decoding for Goodput-Optimized LLM Serving.” *OSDI*. https://arxiv.org/abs/2401.09670.

Zhou, Yanqi et al. 2022. “Mixture-of-Experts with Expert Choice Routing.” *NeurIPS*. https://arxiv.org/abs/2202.09368.

Appendix A — Reproducibility

All projection numbers in §3-5 are regenerated from `figures/generate_figures.py`. Constants and assumptions are in the header of that script. To reproduce:

```
cd paper_02_asic_projection
python figures/generate_figures.py
```

Output: `figures/fig1_radar_inference.png`, `figures/fig2_cost_per_mtokens.png`, `figures/fig3_energy_per_token.png`, `figures/projections_table.csv`.

Appendix B — Changelog

- **v0.3 (2026-04-22)**: Bibliography pass. Added `references.bib` with 31 entries (arXiv IDs, DOIs, vendor URLs). In-text citations converted to pandoc/CSL format. PDF render via `pandoc --pdf-engine=xelatex --citeproc`. Embedded figures (Fig. 1-3) directly in §5. Static landing site (`landing/`) deployable to GitHub Pages / Netlify / Cloudflare Pages.
- **v0.2 (2026-04-22)**: Honesty pass on Table 1 (added chips-per-deployment + per-chip throughput columns). Etched Sohu removed from radar chart, retained in bar charts with unaudited markings. Added bottlenecks B16 (reasoning-tax), B17 (train/inference bifurcation), B18 (numerical determinism). Registry schema v0.2: `estimated_unlock_value`, `depends_on`, `contributors`, `review_cadence_days`. Added Appendix C (agent-density derivation) and §10 (Adversarial Critique).
- **v0.1 (2026-04-22)**: Initial draft. Seed Bottleneck Registry with 15 entries. Projection ranges calibrated to public Groq, Cerebras, NVIDIA MLPerf data.

Appendix C — Agent-density derivation

We define **agent density** D as the number of concurrent Claude-class inference streams sustainable per $1 M of amortized hardware CapEx. Below is the open derivation.

Inputs (Claude-class MoE, ~200-400B active, decode-dominated workload):

Symbol	Meaning
C	CapEx of one node, USD
L	Hardware lifetime, years
u	Average utilization fraction (0-1)
t	tok/s sustained per node, single stream
r	Average tokens/second consumed per active stream (user-facing throughput)

Derivation:

Hourly amortized CapEx per node: `C / (L · 365 · 24)` Cost per million tokens: `C / (L · 365 · 24 · 3600 · u · t) · 1e6` Streams per node: `t / r` Streams per $1 M CapEx: `D = (1e6 / C) · (t / r)`

Worked example — H100 baseline:

```
C  = 30,000 USD
L  = 3 yr
u  = 0.7
t  = 25 tok/s (single-stream Llama-70B BF16, midpoint)
r  = 5 tok/s (representative interactive chat throughput)

streams_per_node       = 25 / 5 = 5
streams_per_$1M        = (1e6 / 30000) · 5 ≈ 167
adjust for utilization = 167 · 0.7 ≈ 117
```

So **D_H100 ≈ 60-150 streams per $1 M**, with the range coming from (a) batch strategy that raises effective t, (b) different r assumptions for verbose vs. terse workloads, (c) utilization across day/night cycles.

Worked example — Reference custom ASIC:

```
C  = 12,000 USD (midpoint of $8-20 K projected)
L  = 3 yr
u  = 0.8
t  = 500 tok/s (midpoint of single-stream projection)
r  = 5 tok/s

streams_per_node       = 500 / 5 = 100
streams_per_$1M        = (1e6 / 12000) · 100 ≈ 8333
adjust for utilization = 8333 · 0.8 ≈ 6667
```

So **D_ASIC ≈ 3,000-12,000 streams per $1 M**, with the range coming from (a) chip cost variance, (b) actual achieved t (depends on FP4 QAT yield), (c) MoE active-parameter assumption.

Sensitivity. D scales linearly in `t / C`. The 50× ratio between H100 and ASIC is dominated by the ~20× throughput improvement and ~2-3× cost reduction. If FP4 QAT delivers only 2× over FP8 (rather than the assumed 4×), `t` halves and D drops to ~1500-6000. Even at this pessimistic case the ratio remains >10×.

Interpretation. D answers a question CEOs and policymakers can act on: "how many always-on Claude agents can a $10 M deployment support?" At H100 economics: ~600-1500. At reference-ASIC economics: ~30,000-120,000. The latter is the regime where "a Claude per knowledge worker" stops being a slogan and becomes a budget line.

---

10. Adversarial Critique (anticipated objections)

A paper of this kind should anticipate the strongest objections and respond before peer review. We list ten.

O1. "You assume FP4 QAT works at frontier scale; nobody has shown that for >1T-parameter MoE." Response: correct; this is bottleneck B4. The projection assumes a 2027 timeline precisely so that FP4 QAT maturation is plausible. We flag this as the dominant single-point risk in §3 and §8.

O2. "Your Claude-class architecture parameters (200-400B active) are speculation." Response: correct, and disclosed in §3.1. Sensitivity analysis (Appendix C) shows D ratios remain >10× even at pessimistic active-parameter assumptions. The conclusion is robust to architecture uncertainty.

O3. "You exclude prefill, which favors GPUs." Response: explicitly excluded in §3.4. Decode is the binding cost in served deployments because it scales with output tokens; prefill is amortized. A full TCO model would include

both; we project decode because it dominates served economics.

O4. "Etched Sohu numbers are not credible." Response: agreed, and v0.2 removes them from the radar chart and explicitly marks them unaudited in tables and bar charts. We retain the row in Table 1 for completeness and as an upper-bound illustration.

O5. "Your agent-density estimate uses an arbitrary `r = 5 tok/s`." Response: yes, this is a representative interactive throughput. Section C derivation is parametric; readers can substitute their own `r`. The relative ratio (D_ASIC / D_H100) is invariant to `r`.

O6. "MLPerf numbers you cite are not for Claude or Anthropic models — they're Llama." Response: correct. Public benchmarks for closed-frontier models do not exist. We use Llama-70B as the closest open analog; this likely underestimates frontier-model cost (frontier models are larger), making our cost-reduction projections conservative.

O7. "The Bottleneck Registry duplicates what SemiAnalysis, AI21, and others already publish." Response: those publish analysis; the registry publishes a *versioned, machine-readable, contribution-friendly* substrate. The novelty is the format and the open-PR mechanism, not the contents of any single entry.

O8. "Your '~80% needs coordination' framing is unfalsifiable." Response: it is testable: each registry entry is tagged by type. A reader can verify the claim by counting entries by type. The taxonomy is explicit (algorithm / architecture / stack / supply / etc.) and can be argued entry by entry.

O9. "You ignore the energy cost of training, which dwarfs inference for frontier labs." Response: scope is explicitly inference (§3.4). Training energy is its own paper-length topic. The argument is that inference economics determine deployment scale, not training economics.

O10. "Software-stack maturity (axis 6 of the radar) is subjective." Response: yes, and disclosed as a 0-10 rubric in §5. We invite competing rubrics. The qualitative finding (dedicated silicon trails CUDA on this axis by 3-6 points) is robust to any reasonable rubric.

Appendix D — Reproducibility manifest (v0.2)

All figures, tables, and projections are regenerated by:

```
cd paper_02_asic_projection
python figures/generate_figures.py
```

Inputs: in-file constants (sources cited inline). Outputs: `figures/fig1_radar_inference.png` (5 platforms, Etched excluded), `fig2_cost_per_mtokens.png` (6 platforms, Etched marked unaudited), `fig3_energy_per_token.png` (same), `projections_table.csv` (all 6 with ranges).

Registry: `bottleneck_registry.md` (18 entries, v0.2 schema for B16-B18, v0.1 schema for B1-B15).