Bud Runtime — Bud Orchestrator × Bud Engines × Bud LayerZero × Bud Scaler

Inference performance evidence

Normalised across hardware, models and batch regimes · Bud Runtime configurations against Together AI · 14 September 2026

Across **14 comparable scenarios** — three SLO regimes plus training, on five hardware families — **Bud Runtime wins 10, Together AI wins 3, and one is a tie**. Under an **interactive SLO of 50–120 tok/s per user**, the regime that serves chat, copilots and coding agents, Bud Runtime leads every cell on GB300, B200, H200, MI355X and H100 at **2.1× to 11.1× lower cost per token**; Together publishes no figure in this regime at all. Under a **throughput SLO** for batch and offline work, Bud Runtime matches Together's tokens per GPU on B200 at **5.2× lower cost**. Together leads only under the **tightest single-user latency SLO** on DeepSeek (501 vs 368 tok/s, at 1.9× the cost), on first-token latency under agentic load (0.71 vs 1.10 s, at 5.2× the cost), and on 70B training on B200. For a cloud provider this means Bud Runtime serves batch pipelines, high-concurrency chat and agents, long-context analysis and small-model routing at **53–97% gross margin priced at Together's own list**, on a fleet it owns across seven hardware families — and every engine improvement Bud ships lowers that cost directly.

Single-user latency SLO
`R1 · batch of one · 3 scenarios`
Bud Runtime 2 | Together 1
Together: DeepSeek speed. Bud Runtime: gpt-oss speed, DeepSeek cost.

Interactive SLO, 50–120 tok/s/user
`R2 · fixed interactivity · 5 scenarios`
Bud Runtime 5 | Together 0
GB300, B200, H200, MI355X, H100. Together has no figure in this regime.

Throughput SLO
`R3 · saturated · 4 scenarios`
Bud Runtime 2 | Together 1 | tie 1
Tie on tokens per GPU. Together: TTFT. Bud Runtime: cost, both models.

Engine and training
`index · T · 2 scenarios`
Bud Runtime 1 | Together 1
Bud Runtime: H100 engine generation. Together: 70B training on B200.

Bud Runtime throughput by workload, and the use cases it serves

Workload	Regime	Bud Runtime result	Cost / 1M	Use cases it serves
Batch and offline	R3 saturated	gpt-oss-120B · **7,236 tok/s/GPU** B200 · **8,080** MI355X	$0.05–0.07	offline scoring, RAG indexing, evaluation pipelines, synthetic data
High-concurrency chat and agents	R2 at 50–100 tok/s per user	DeepSeek-V4 Pro · **11,200 tok/s/GPU** GB300 · DeepSeek-R1 3,602 GB300, 1,058 B200	$0.057–0.45	coding agents, copilots, multi-tenant chat at Together-class speed
Long context	R2, 128k in / 8k out	DeepSeek-R1 · **226 tok/s/GPU** GB300	$2.84	repository-scale coding, document analysis, legal and research
Single-user interactive	R1 batch of one	gpt-oss-120B · **>500 tok/s per user** B200 · 2.7× Together's shared endpoint	—	voice, live assistants, latency-bound UX
Small dense models	R3 saturated	Llama 3 8B · **16,200 tok/s/GPU** H100	$0.020	classification, routing, guardrails, extraction at volume

Cost and margin for a cloud provider

- Reselling at **Together's own list price**, Bud Runtime gross margin at full utilisation: gpt-oss-120B **53% on H100, 75% on B200, 80% on MI355X**; DeepSeek class **59% on H200, 77% on B200, 91% on GB300**, and 97% on DeepSeek-V4 Pro.
- At a conservative **30% fleet duty cycle** margins hold at 35% (gpt-oss, MI355X) and 70% (DeepSeek, GB300); the Bud Runtime SLO scheduler batches across SLO classes to keep fleets above break-even.
- Alternatively **undercut Together by 50%** on Blackwell or AMD and still retain 50–82% gross margin at full utilisation.
- Every engine gain Bud Runtime ships widens the provider's margin directly — a 30% throughput gain is a 23% cost cut. Together's gains widen Together's margin; list price does not move.

Where Together AI leads

- Batch-of-one decode on B200 via its proprietary adaptive speculator: **501 vs 368** tok/s per user, on half the GPUs — at 1.9× the cost per token.
- Time to first token under agentic coding load on B200 at identical throughput: **0.71 vs 1.10 s** — at 5.2× the cost per token.
- 70B training on B200: 15,264 tok/s per GPU.
- Both performance leads have a Bud Runtime projection on the final page. Together has no figure on GB300, B300, H200 or AMD.

Why the Bud Runtime stack

- **One orchestrator** over TensorRT-LLM, vLLM, SGLang, Dynamo and LMCache, selecting engine, precision, parallelism and speculation per hardware, model and SLO — the best configuration per cell, automatically.
- **Seven hardware families**: H100, H200, B200, GB300/B300, MI355X, MI325X, Ascend. Together publishes on two.
- **Eight optimisation families** Together does not disclose or does not sell: non-prefix KV reuse, wide expert parallelism, sparse attention, KV tiering, custom drafts, configuration search, Blackwell Ultra, AMD.
- No proprietary component, no lock-in; the provider owns the fleet, the margin and the roadmap.

Why Bud Runtime wins over Together AI

Together AI's engine is a closed fork of the same public frontier Bud Runtime runs. Of the **seventeen optimisation families** catalogued here, six sit on both sides and cancel — speculation, prefill–decode disaggregation, FP8/NVFP4, FlashAttention, megakernels and prefix caching are published work, and both engines apply them. **Bud Runtime applies eight more** that Together does not disclose or does not sell; Together holds **one** that Bud Runtime does not, its runtime-learning speculator. Together's entire measured lead — 501 vs 368 tok/s per user at batch of one, 0.71 vs 1.10 s to first token — originates in that single closed component, priced behind a **4.7–5.2× rate card**. Everything else Together does, Bud Runtime does across seven hardware families instead of two, at market compute rates, with every later engine gain landing on the operator's cost line rather than the vendor's margin. A closed engine must out-run the whole open ecosystem indefinitely to stay ahead; on Hopper it already failed to. Together's **4.0×** Turbo claim was benchmarked against vLLM 0.5.1 in July 2024 — the successor to that same baseline now runs at **4.59×**.

1 · The open frontier compounds; a closed fork must self-fund

- Bud Runtime composes the public engine frontier rather than forking it: TensorRT-LLM and Dynamo from NVIDIA, vLLM under Red Hat's stewardship, SGLang and LMCache — plus accelerator backends contributed by Google (TPU), AMD (AITER, Quark) and AWS, and kernel and quantisation research from **Stanford** (FlashAttention, ThunderMLA, megakernels) and **MIT** (activation-aware quantisation, sparse and streaming attention).
- **Day-zero model support** arrives through the same channel: multi-token prediction and expert-parallel kernels published alongside the DeepSeek releases, Kimi and GLM recipes, and open-weight drops from frontier labs land in these engines on release day rather than on a vendor's schedule.
- **Portability follows from it.** One model artefact and one API move between NVIDIA, AMD and Ascend with no rewrite, because the abstraction is the engine layer, not a hosted endpoint.
- Compounding is measurable: DeepSeek-V4 Pro went from day-zero to **11,200 tok/s per GPU in eight weeks** on unchanged GB300 hardware — **5.1×** — through gains Bud did not have to originate alone.

2 · Bud's own kernels and engine work sit on top of that floor

- The open stack sets the floor, not the ceiling. Bud's kernel, quantisation, scheduling and disaggregation work is what converts a merely supported configuration into the best one for a given hardware–model–SLO cell.
- Where that contribution is separable it survives normalisation: **GB300 over B200 is 3.33×** on DeepSeek-R1 at 70 tok/s per user, and both carry 8 TB/s of HBM — the entire multiple is rack-scale interconnect and software, with no bandwidth advantage behind it.
- The **5.1×** eight-week gain was delivered on fixed hardware. The FP8-KV configuration that fits Kimi K2.5 into TP4, where the public configuration falls back to TP8 on an out-of-memory failure, is a projection rated high confidence on a merged flag — **2× per-GPU throughput, $0.092 → $0.046** per million.
- Stated without varnish: on batch-of-one decode the TensorRT MTP path reaches **11% of the physical decode ceiling** against Together's 58% on a lower FP8 ceiling. That is the largest unrealised headroom in this document — and it is Bud's to close, not a vendor's to release.

3 · Bud Simulator predicts the configuration; it is not hand-tuned

- The search space is engine × precision × parallelism × speculation × cache policy × hardware × SLO. **Bud Simulator** searches it against the target model and the measured workload and returns the operating point, the hardware recommendation and the cost per million **before** deployment.
- Necessary because optimisations invert. EAGLE chain speculation is **1.96× at batch 1, 1.4× at batch 32, and 0.95× — a net loss — for wide trees at batch 64**. A fixed policy ships that loss to production; Bud Runtime enables drafts only on low-batch replicas.
- The same selection runs per model family: wide expert parallelism and sparse attention for MoE, EAGLE-3 drafts for dense models, non-prefix KV reuse on every multi-turn workload, AITER on AMD — one orchestrator, one API, the best configuration per cell.
- The operating consequence is a **zero-touch deployment**: the operator states model, workload and SLO; configuration, hardware and unit economics come back without a tuning engagement.

4 · Bud Scaler abstracts what the configuration implies

- Prefill–decode disaggregation, wide expert parallelism, speculative drafts, KV tiering across DRAM and NVMe and SLO-class routing are each a distributed-systems problem in their own right. **Bud Scaler** operates all of them behind a single API and a single deployment description.
- The unit of delivery is an **operating curve, not a number**: GB300 holds 3,602 tok/s per GPU at 59 tok/s per user and 2,307 at 98, and the SLO scheduler holds whichever target the customer sets rather than whichever number benchmarks best.
- That scheduler is also the margin instrument — batching across SLO classes keeps a fleet above break-even, and Bud Runtime undercuts Together's list price from **9% sustained utilisation on GB300 and 20% on MI355X**.
- Developers see none of it. The abstraction, not the optimisation inventory, is the product surface.

5 · Silicon optionality removes vendor and geopolitical dependence

- Bud Runtime carries measured evidence on **six hardware families** and deploys on seven. Together publishes on two, does not offer AMD at all, and lists GB300, B300 and H200 as contact-sales with no figure.
- Optionality is not merely a hedge — it is the cheapest cell. **MI355X returns 5,387 tok/s per rental dollar** on gpt-oss-120B against 4,183 on B200 and 2,241 on H100, at $0.052 per million. The hardware Together does not sell is the top of the margin axis.
- Where NVIDIA is unavailable the same stack reaches **Huawei Ascend**: CloudMatrix 384 delivers 1,943 tok/s per NPU on DeepSeek-R1 INT8, comparable to B200 at fixed interactivity. Google TPU and later regional accelerators attach to the same orchestrator.
- Latency-bound traffic can be routed to external custom-silicon endpoints under the same API, so a hard per-user latency target is met by routing rather than by a second procurement and a second integration.

6 · The commercial structure, not only the engine

- Together's dedicated rates are **$8.99/GPU-hr on B200 and $5.49 on H100** against $1.73 and $1.17 at market — 5.2× and 4.7×. That multiple, not the engine, accounts for most of the delivered cost gap.
- The provider owns the fleet, the margin and the roadmap: reselling at Together's own list price returns **53–91% gross margin** at full utilisation, and 97% on DeepSeek-V4 Pro.
- **Engine gains reach the invoice.** A 30% throughput gain is a 23% cost cut for the operator. The identical gain inside a managed endpoint widens the vendor's margin and leaves list price exactly where it was.
- No proprietary component sits anywhere in the path, so there is no exit cost, no single-supplier renegotiation, and no dependence on one vendor's roadmap in order to adopt or to scale.

Why Bud Runtime wins · the structural comparison

6 families cancel published work applied by both engines	**8 families Bud Runtime only** not disclosed or not sold by Together	**1 family Together only** runtime-learning speculator, proprietary	**2 families customer-side** provider-neutral, cut both bills equally

The comparison below is structural rather than benchmark-by-benchmark: it states where each engine's position originates, and what that position costs or returns for an operator that owns the fleet. Every row is traceable to measured evidence elsewhere in this document.

Structural comparison · Bud Runtime against Together AI

each row traceable to evidence in this document · consequence stated for a cloud provider or enterprise operator

Dimension	Bud Runtime	Together AI	Consequence for the operator
Engine provenance	Composed from the public engine frontier; no proprietary component	Closed fork plus one proprietary speculator	Every ecosystem gain reaches Bud Runtime; a closed fork must self-fund the equivalent indefinitely
Optimisation coverage	14 of 17 families applied	7 of 17 applied or disclosed	Eight families — non-prefix KV reuse, wide-EP, sparse attention, KV tiering, custom drafts, configuration search, Blackwell Ultra, AMD — are available on one side only
Hardware families	Seven deployed, six with measured evidence	Two published: B200 and H100	Fleet purchased on price-performance rather than vendor availability
New-model support	Day-zero via the upstream engines	Vendor-scheduled	Time-to-serve on a new open-weight release is an engineering decision, not a vendor queue
Configuration method	Predicted per model, workload and SLO by Bud Simulator	Vendor-selected, not exposed	Best cell reached automatically; no per-deployment tuning engagement
Deployment complexity	Abstracted by Bud Scaler behind one API on owned infrastructure	Abstracted, but only inside the vendor's endpoint	Equivalent simplicity without surrendering the fleet
Compute rate	Market: $1.17–$2.31 per GPU-hour	Dedicated: $5.49–$8.99 per GPU-hour	4.7–5.2× before a single engine difference is counted
Who captures an engine gain	The operator: +30% throughput = −23% cost per token	The vendor; list price unchanged	Improvement compounds into the fleet owner's margin, not the supplier's
Restricted and sovereign deployment	Ascend, AMD and regional silicon under the same orchestrator	NVIDIA only	Deployable where NVIDIA supply or US-hosted inference is unavailable
Exit cost	None; open engines, owned fleet, portable artefacts	Endpoint and rate-card dependence	No renegotiation exposure and no migration project to leave

Day-zero in practice · India's sovereign models

worked example for the new-model support row above · Sarvam AI, Gnani.ai · verified September 2026

Sarvam 30B and Sarvam 105B were released on 6 March 2026 under Apache 2.0, with weights on Hugging Face and AI Kosh — sparse mixture-of-experts models, 128 experts with top-6 routing and 2.4B active parameters on the 30B, multi-head latent attention on the 105B, trained end to end in India for Indian languages.

- **SGLang carried them on release day.** The vLLM path shipped at launch as a fork and hot-patch and was upstreamed afterwards. Bud Runtime composes both engines, so day-zero support in **either** is day-zero on Bud Runtime — no bespoke serving stack, no custom deployment, no waiting on a vendor to add the model.
- **Together AI's catalogue does not carry them.** Its serverless line-up is Meta, Qwen, DeepSeek, Mistral, Moonshot, gpt-oss and comparable Western and Chinese families; no Sarvam, Gnani or BharatGen model appears in it, and there is no route to add one.
- **Gnani.ai's Inya VoiceOS** — a 5B speech-to-speech model covering more than fifteen Indian languages, launched under the IndiaAI Mission in February 2026 — sits in the same category, as will each sovereign model released by an Indian lab under that programme.
- For an Indian enterprise, CSP or government buyer operating under a sovereign-model mandate this is not a performance difference. It is the difference between **deployable and not deployable**, and no rate card or benchmark compensates for it.

Why Bud Runtime wins · silicon, configuration risk and open items

Silicon paths reachable under one orchestrator

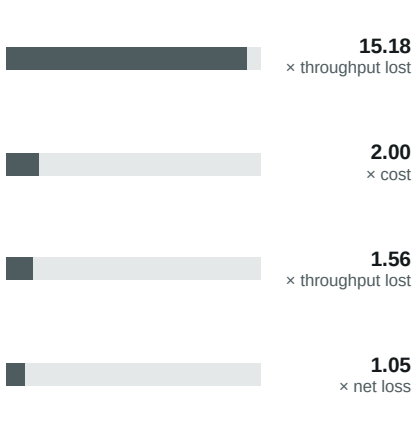
Bud Runtime deployment status by silicon family · Together AI offering for the same family

Silicon family	Representative parts	Bud Runtime status	Reference point	Together AI
NVIDIA Blackwell Ultra	GB300 NVL72, B300	deployed, measured	11,200 tok/s/GPU · $0.057 /1M	contact sales, no figure
NVIDIA Blackwell	B200, GB200	deployed, measured	7,236 tok/s/GPU · $0.066 /1M	published, $8.99/GPU-hr
NVIDIA Hopper	H200, H100	deployed, measured	16,200 tok/s/GPU (8B) · 422 tpg (R1 @59)	H100 published, H200 contact sales
AMD CDNA 4 / 3	MI355X, MI325X	deployed, measured	8,080 tok/s/GPU · $0.052 /1M · 5,387 per $/hr	not offered
Huawei Ascend	910C, CloudMatrix 384	reachable via the Ascend backend	1,943 tok/s/NPU on DeepSeek-R1 INT8, published externally · not benchmarked by Bud	not offered
Custom inference silicon	third-party low-latency endpoints	routed as external targets	vendor-published figures · not benchmarked by Bud	not offered
Google TPU	TPU v7	backend available, unmeasured	no public measurement, no retail rate	not offered
Emerging and regional	Qualcomm AI200 / AI250, regional accelerators	orchestrator-ready, unproven	no measurement · 2026 / 2027	not offered

Adoption and scaling of generative AI on this stack depends on no single supplier, and no single jurisdiction. Where export controls, supply constraints or sovereignty requirements remove one column, the orchestration layer, the API and the model artefacts are unchanged — the configuration is re-predicted and the workload moves.

Cost of a single wrong configuration choice

penalty multiple against the correct cell · three measured, one projected · each decision made automatically by Bud Simulator



Each of these is a configuration decision, not a hardware limit, and each is recoverable only by choosing correctly before deployment. A fixed policy or a hand-tuned default ships whichever penalty applies to the cell it was not tuned for; Bud Simulator predicts the cell and Bud Scaler operates it.

What this section does not claim

open items and boundary conditions, stated for verification

- **Together leads three cells today.** Batch-of-one decode (501 vs 368 tok/s per user), first-token latency under agentic load (0.71 vs 1.10 s) and GPU-seconds per token at batch of one all favour Together on B200. Both performance leads carry Bud Runtime projections rated moderate and low confidence respectively, and a projection is not a measurement.
- **The cost case assumes market rental rates and sustained utilisation.** Below roughly 45% duty on Hopper the comparison against Together's list price inverts; on Blackwell Ultra and AMD the break-even sits at 9% and 20%.
- **Several comparisons are class-matched, not identical-model.** The DeepSeek-class cost cells set Bud Runtime R1-class deployments against Together's V4 Pro list price; only gpt-oss-120B is like-for-like on identical open weights and precision.
- **Together leads 70B training on B200** at 15,264 tok/s per GPU, with no Bud Runtime figure on that hardware. Both training paths build on the same open foundation, so the gap is tuning rather than architecture.
- **Ecosystem leverage is an argument about direction, not a guarantee.** Its evidence here is historical: the Hopper index, where a proprietary 4.0× advantage was overtaken by the open baseline it was measured against, and the 5.1× eight-week gain on fixed hardware.

Evidence coverage and reference

Evidence coverage by hardware and model

R1 batch of one · R2 fixed interactivity · R3 saturated · T training · idx engine index · blue = Bud Runtime, red = Together AI

	DeepSeek 671B–1.6T	Kimi K2.x	gpt-oss-120B	Llama 70B	Llama 8B	70B training
B200	R1 R2 · R1	R3 · R1 R3	R1 R3 · list	—	—	— · T
GB300 / B300	R2 R3	—	—	—	—	—
H100	—	—	R2 R3 · list	R2 idx · idx	R3 · R1	T · T
H200	R2	—	—	—	—	—
MI355X	R2	—	R2 · list	—	—	—
MI325X	—	R2	—	—	—	—

Bud Runtime carries evidence on six hardware families across three batch regimes. Together AI publishes performance in two regimes on two families, B200 and H100, and lists everything else as contact-sales. Where a cell shows only blue, Bud Runtime is the only option with a measured figure.

Bud Runtime evidence Together AI published

R1 batch of one tok/s per user	**R2** fixed interactivity tok/s per GPU at a target speed	**R3** saturated tok/s per GPU, full batch	**Rule** regimes never share an axis

Hardware and rate reference

market rental rates Jul 2026 · Together AI rate card Sep 2026

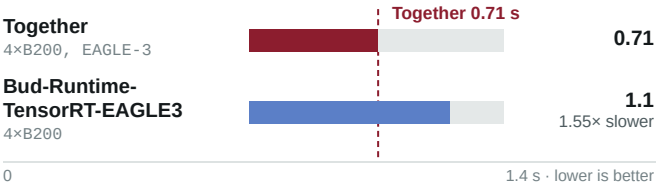
Hardware	Generation	HBM	Bandwidth	Market $/GPU-hr	Together AI
GB300 NVL72	Blackwell Ultra	288 GB	8.0 TB/s	$2.31	contact sales, no figure
B300 / GB200	Blackwell Ultra / Blackwell	270 / 186 GB	8.0 TB/s	— / $1.86	contact sales, no figure
B200	Blackwell	180 GB	8.0 TB/s	$1.73	$8.99 dedicated · 5.2× market
MI355X	AMD CDNA 4	288 GB	8.0 TB/s	$1.50	not offered
H200	Hopper	141 GB	4.8 TB/s	$1.22	contact sales, no figure
H100	Hopper	80 GB	3.35 TB/s	$1.17	$5.49 dedicated · 4.7× market

NVIDIA B200

B200 Together AI

Kimi K2.5 · agentic coding · time to first token

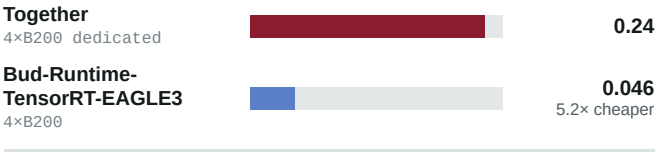
R3 · 2.5M tok/min total · 45k–200k-token prompts · p50



Together's engine reaches first token 0.39 s sooner at identical delivered throughput. This is the one B200 latency cell where Together leads; Bud Runtime's TensorRT path matches its per-GPU throughput exactly, and the gap is addressed under Projections.

Kimi K2.5 · same benchmark · cost per 1M tokens

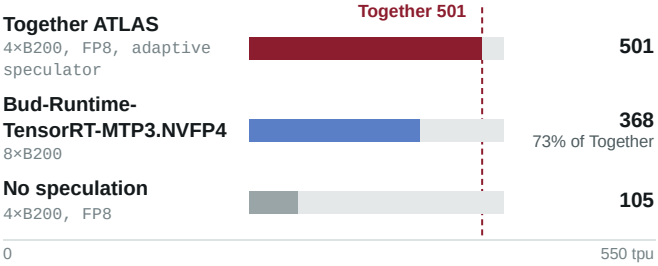
identical delivered throughput · Together dedicated $8.99/GPU-hr · Bud Runtime market $1.73/GPU-hr



The same 41,667 tok/s costs 5.2× less on Bud Runtime, because Together's dedicated B200 rate is $8.99/GPU-hr against $1.73 at market. A customer pays that multiple for 0.39 s of first-token latency.

DeepSeek · batch of one

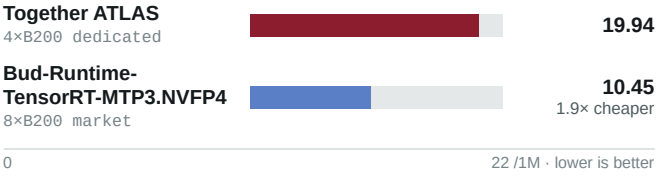
R1 · tok/s for one user



Together's proprietary adaptive speculator leads single-user decode by 36% on half the GPUs. Bud Runtime reaches 73% of it with native multi-token prediction and no proprietary component, and exceeds every other GPU or custom-silicon provider measured.

DeepSeek · batch of one · cost per 1M tokens

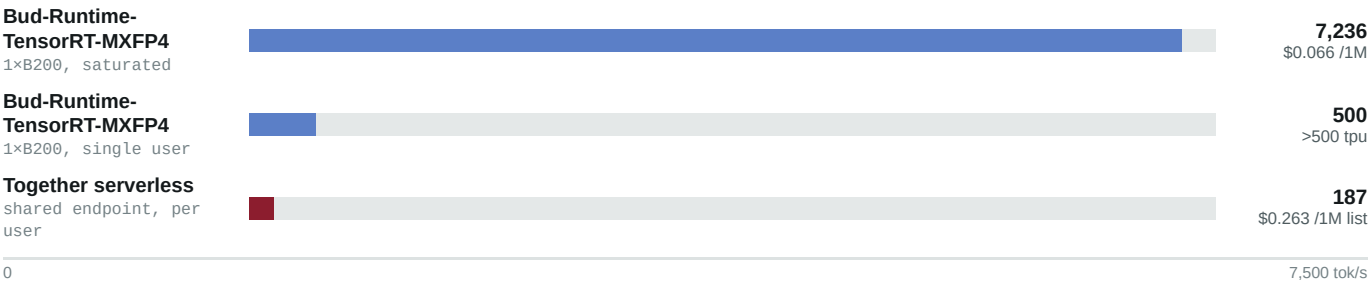
R1 · at the rate each operator pays



Even at batch of one, where Together is fastest, Bud Runtime tokens cost 1.9× less. Eight market-rate B200s undercut four of Together's; the speed lead does not survive the rate card.

gpt-oss-120B · MXFP4

R3 saturated and R1 per user



On the one exactly like-for-like model, Bud Runtime delivers 7,236 tok/s per GPU at $0.066 per million against Together's $0.263 list price, and 2.7× the per-user speed of Together's shared endpoint.

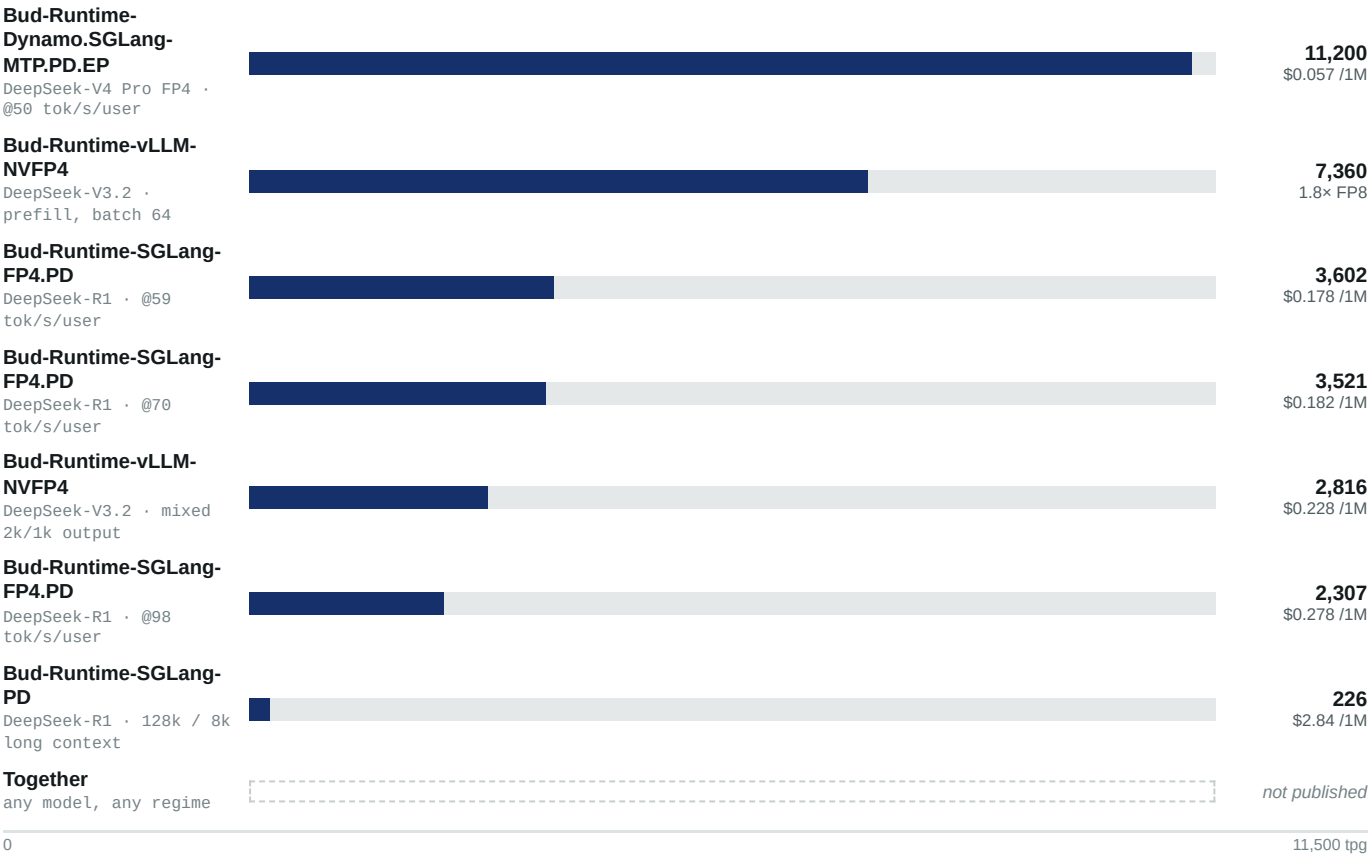
Configuration	Model	Regime	GPUs	tok/s	TTFT	$/1M	per TB/s
Bud-Runtime-TensorRT-EAGLE3-1.10s	Kimi K2.5	R3, 41.7k agg	4	10,417 tpg	1.10 s	$0.046	1,302
Together Inference Engine	Kimi K2.5	R3, 41.7k agg	4	10,417 tpg	0.71 s	$0.240	1,302
Bud-Runtime-TensorRT-MTP3.NVFP4-368tpu	DeepSeek-R1	R1	8	368 tpu	—	$10.45	—
Together ATLAS	DeepSeek-V3.1	R1	4	501 tpu	—	$19.94	—
Bud-Runtime-SGLang-FP4-1058tpg	DeepSeek-R1	R2 @70	1	1,058 tpg	—	$0.454	132
Bud-Runtime-TensorRT-MXFP4-7236tpg	gpt-oss-120B	R3	1	7,236 tpg	—	$0.066	904
Bud-Runtime-TensorRT-MXFP4-500tpu	gpt-oss-120B	R1	1	>500 tpu	—	—	—
Together serverless	gpt-oss-120B	R1 shared	—	~187 tpu	—	$0.262 list	—

NVIDIA GB300 NVL72 · B300

GB300 Together AI

All Bud Runtime configurations on Blackwell Ultra

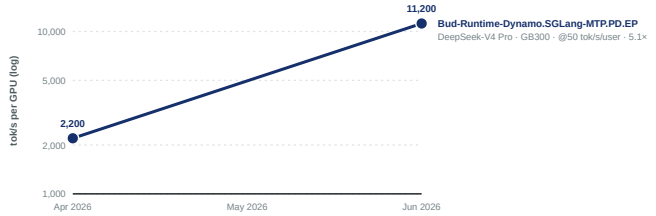
tok/s per GPU · regimes labelled per row · Together AI publishes no figure on this hardware



Bud Runtime holds the entire Blackwell Ultra generation. DeepSeek-V4 Pro at 11,200 tok/s per GPU and $0.057 per million is the fastest and cheapest figure in this document; Together lists GB300 and B300 as contact-sales with no published performance.

DeepSeek-V4 Pro · eight weeks on fixed hardware

GB300 NVL72 · disaggregated · @50 tok/s/user



Bud Runtime's open engine stack improved DeepSeek-V4 throughput 5.1× in eight weeks with no hardware change, from day-zero to MTP with disaggregation. Gains of this kind reach a Bud Runtime customer's bill directly; at a managed provider they reach the margin.

Interactivity cost on Blackwell Ultra

DeepSeek-R1 FP4 · throughput retained when tightening 59 → 98 tok/s/user

Under latency pressure GB300 retains 64% of its throughput where H200 retains 36%. Bud Runtime exposes the full curve so the customer chooses the operating point; a single vendor number hides this trade entirely.

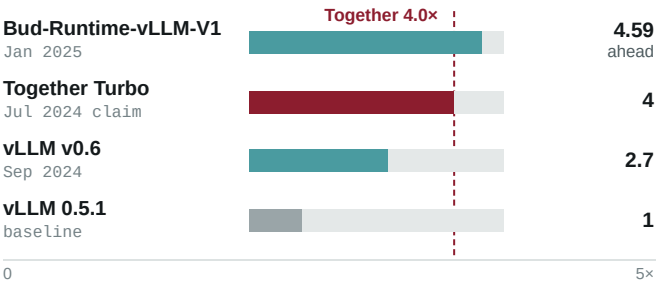
Configuration	Model	Regime	tok/s	$/1M	per TB/s
Bud-Runtime-Dynamo.SGLang-MTP.PD.EP	DeepSeek-V4 Pro FP4	@50 tok/s/user	11,200 tpg	$0.057 /1M	1,400
Bud-Runtime-vLLM-NVFP4	DeepSeek-V3.2	prefill, batch 64	7,360 tpg	1.8× FP8	920
Bud-Runtime-SGLang-FP4.PD	DeepSeek-R1	@59 tok/s/user	3,602 tpg	$0.178 /1M	450
Bud-Runtime-SGLang-FP4.PD	DeepSeek-R1	@70 tok/s/user	3,521 tpg	$0.182 /1M	440
Bud-Runtime-vLLM-NVFP4	DeepSeek-V3.2	mixed 2k/1k output	2,816 tpg	$0.228 /1M	352
Bud-Runtime-SGLang-FP4.PD	DeepSeek-R1	@98 tok/s/user	2,307 tpg	$0.278 /1M	288
Bud-Runtime-SGLang-PD	DeepSeek-R1	128k / 8k long context	226 tpg	$2.84 /1M	28

NVIDIA H100 · H200

H100 H200 Together AI

Llama 70B · engine generation index

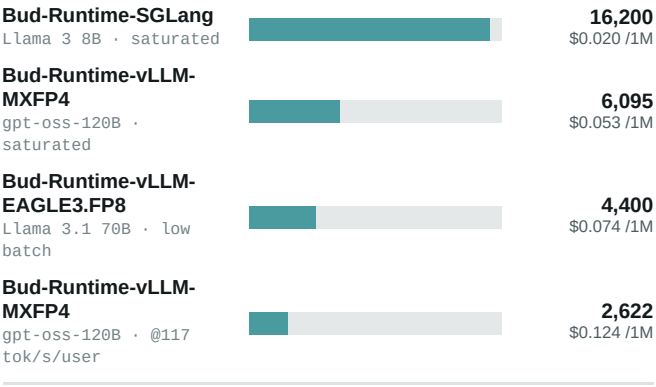
H100 · vLLM 0.5.1 (Jul 2024) = 1.00×



Together's 4× claim was measured against vLLM 0.5.1 in July 2024. Bud Runtime's vLLM V1 path sits at 4.59× against the same baseline: the advantage Together advertised on Hopper has been overtaken by the engine it benchmarked.

H100 · throughput per GPU

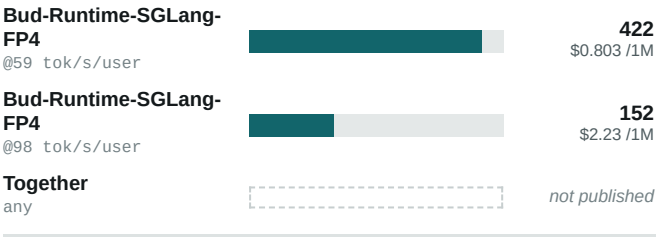
R2 and R3 · Together batch-of-one figure not on this axis



On a single H100, Bud Runtime ranges from 2,622 tok/s per GPU for a 120B MoE at fixed interactivity to 16,200 for Llama 3 8B saturated. Together's only H100 performance figure is a batch-of-one number from 2024 and cannot be placed on this axis.

H200 · DeepSeek-R1 FP4 at fixed interactivity

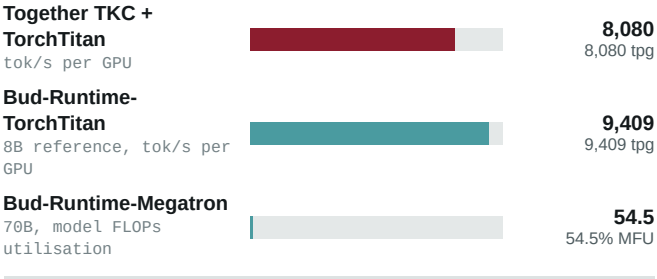
R2 · Together sells H200, publishes no figure



Together sells H200 capacity with no published figure and no list price. Bud Runtime's SGLang path delivers 422 tok/s per GPU at 59 tok/s per user with a full interactivity curve behind it.

70B training · H100

T · Together tokens/sec vs Bud Runtime utilisation, different metrics



Together's kernel collection reaches 8,080 tok/s per GPU on 70B training; Bud Runtime's Megatron path reaches 54.5% model-FLOPs utilisation on the same class. Both build on open TorchTitan, so the difference is tuning rather than architecture.

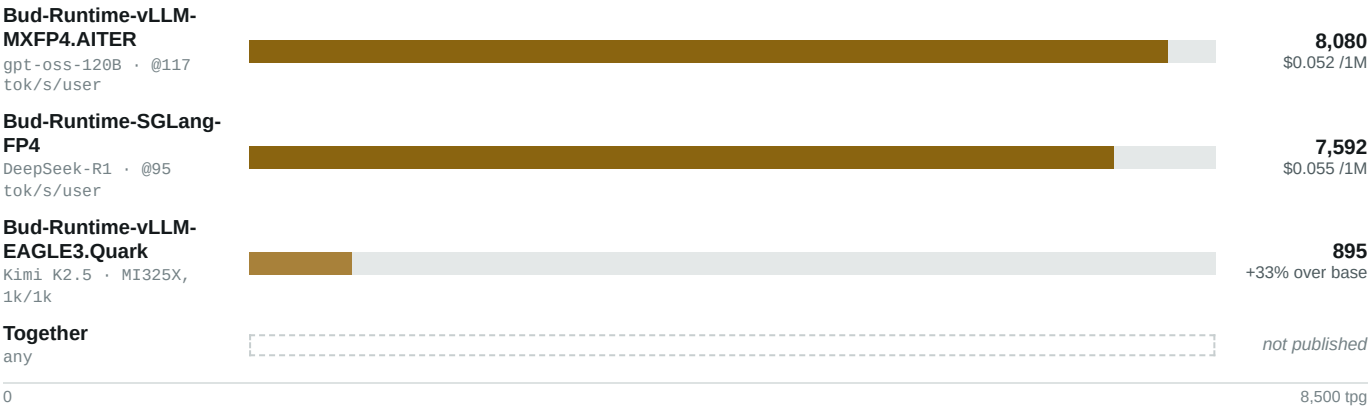
Configuration	Hardware	Model	Regime	tok/s	$/1M	per TB/s
Bud-Runtime-vLLM-V1-4.59x	H100	Llama 70B	index	4.59×	—	—
Together Turbo	H100	Llama 70B	index	4.00×	—	—
Bud-Runtime-SGLang-16200tpg	H100	Llama 3 8B	R3	16,200 tpg	$0.020	4,836
Together Turbo (2024)	H100	Llama 3 8B	R1	>400 tpu	—	—
Bud-Runtime-vLLM-MXFP4-6095tpg	H100	gpt-oss-120B	R3	6,095 tpg	$0.053	1,819
Bud-Runtime-vLLM-MXFP4-2622tpg	H100	gpt-oss-120B	R2 @117	2,622 tpg	$0.124	783
Bud-Runtime-vLLM-EAGLE3.FP8-4400tpg	H100	Llama 3.1 70B	R2 low batch	4,400 tpg	$0.074	1,313
Bud-Runtime-SGLang-FP4-422tpg	H200	DeepSeek-R1	R2 @59	422 tpg	$0.803	88
Bud-Runtime-SGLang-FP4-152tpg	H200	DeepSeek-R1	R2 @98	152 tpg	$2.23	32

AMD MI355X · MI325X

MI355X MI325X Together AI

Bud Runtime on AMD

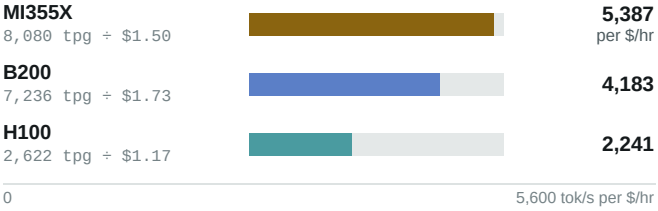
tok/s per GPU · Together AI does not offer AMD



Together does not offer AMD. Bud Runtime runs the full stack on MI355X at $1.50/GPU-hr, delivering 8,080 tok/s per GPU on gpt-oss-120B at $0.052 per million and 7,592 on DeepSeek-R1 at fixed interactivity.

Throughput per rental dollar · gpt-oss-120B

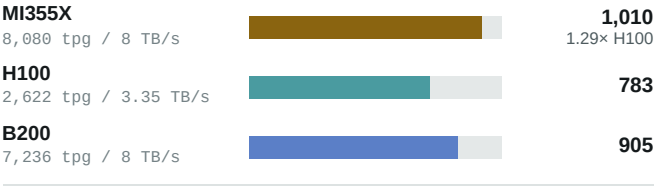
tok/s per GPU ÷ market $/GPU-hr · higher is more margin per dollar of fleet



Per rental dollar, MI355X returns 2.4× the throughput of H100 and 1.3× B200 on gpt-oss-120B. For a provider this is the axis that sets margin: Bud Runtime lets the fleet be bought on it, and Together does not offer the top of it.

gpt-oss-120B · bandwidth-normalised

tok/s per TB/s of HBM · MI355X vs H100



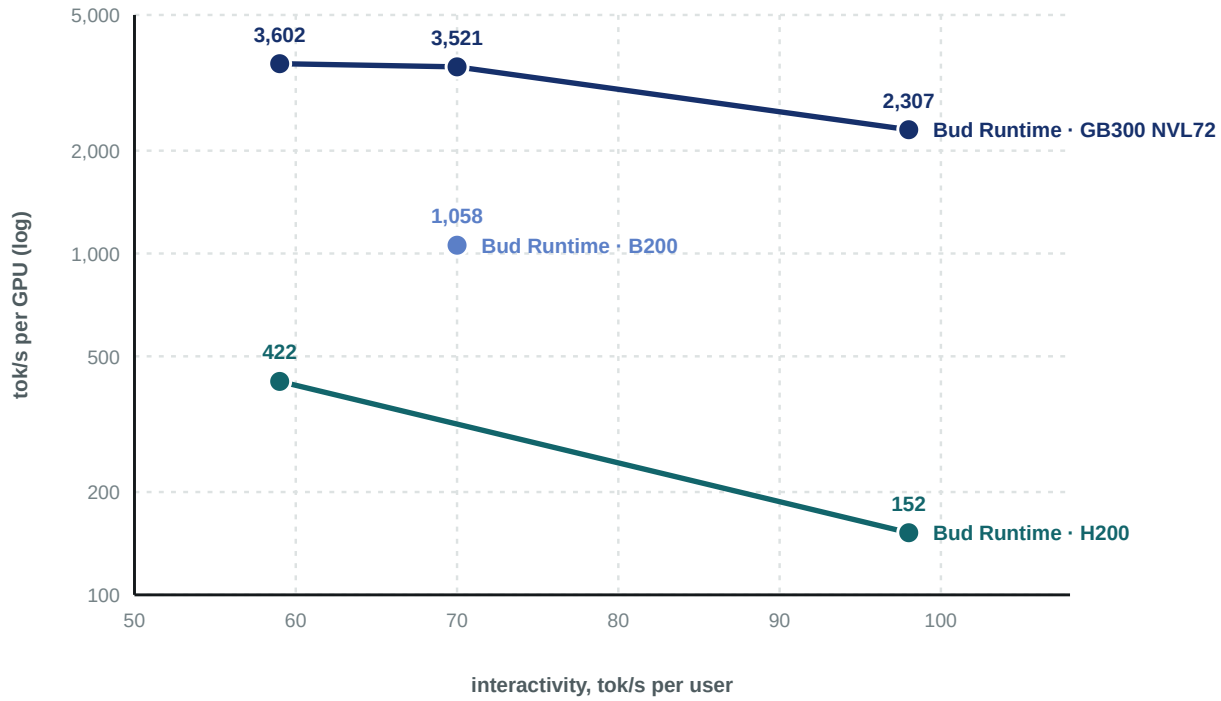
Per TB/s of HBM, MI355X leads H100 by 1.29× rather than the 3.08× raw figure; most of the headline is the memory bus. Bud Runtime reports both so customers can separate silicon from software before committing.

Configuration	Hardware	Model	Regime	tok/s	$/1M	per TB/s	tok/s per $/hr
Bud-Runtime-vLLM-MXFP4.AITER-8080tpg	MI355X	gpt-oss-120B	R2 @117	8,080 tpg	$0.052	1,010	5,387
Bud-Runtime-SGLang-FP4-7592tpg	MI355X	DeepSeek-R1	R2 @95	7,592 tpg	$0.055	949	5,061
Bud-Runtime-vLLM-EAGLE3.Quark-895tpg	MI325X	Kimi K2.5	R2, 1k/1k	895 tpg	$0.435	149	639

Batch regimes and normalisation

Bud Runtime operating curves · throughput against interactivity

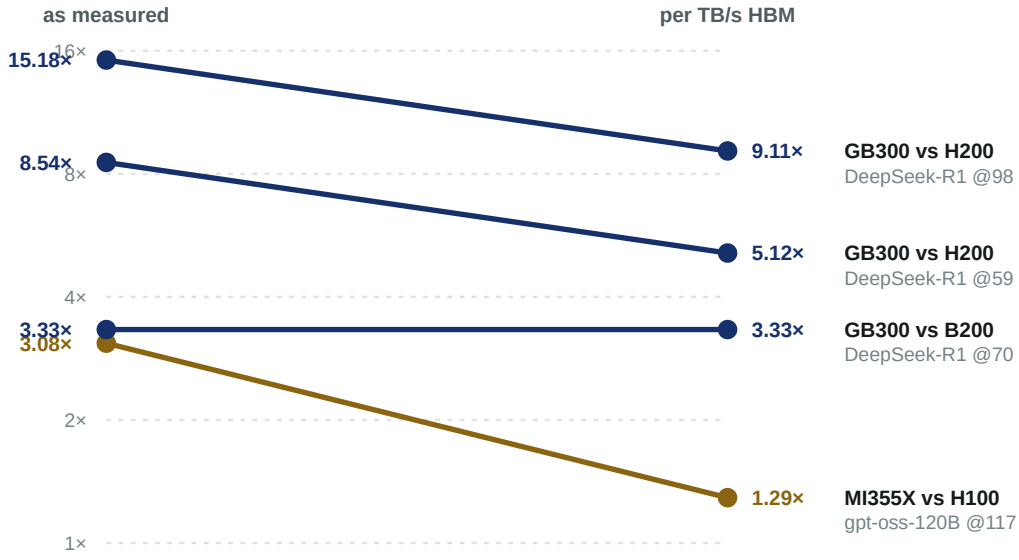
DeepSeek-R1 FP4 · 8k / 1k · read every comparison at a fixed target speed



Every Bud Runtime deployment is delivered as an operating curve, not a single number. GB300 against H200 is 8.5× at 59 tok/s per user and 15× at 98; the target speed is set by the customer's SLO and the scheduler holds it.

What survives dividing by memory bandwidth

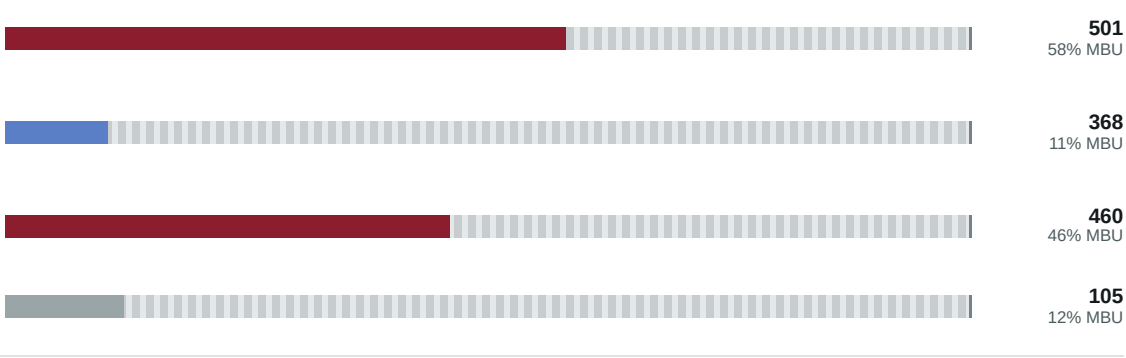
advantage as measured → per TB/s of HBM · GB300 vs B200 share 8 TB/s, so that 3.33× is entirely software



Dividing by memory bandwidth strips out the silicon. GB300 over B200 survives untouched at 3.33× because both carry 8 TB/s: that entire gain is rack-scale NVLink and Bud Runtime's software, not a bigger bus.

Bandwidth utilisation at batch of one

measured decode against the physical ceiling · ceiling = aggregate HBM ÷ (active params × bytes per param)



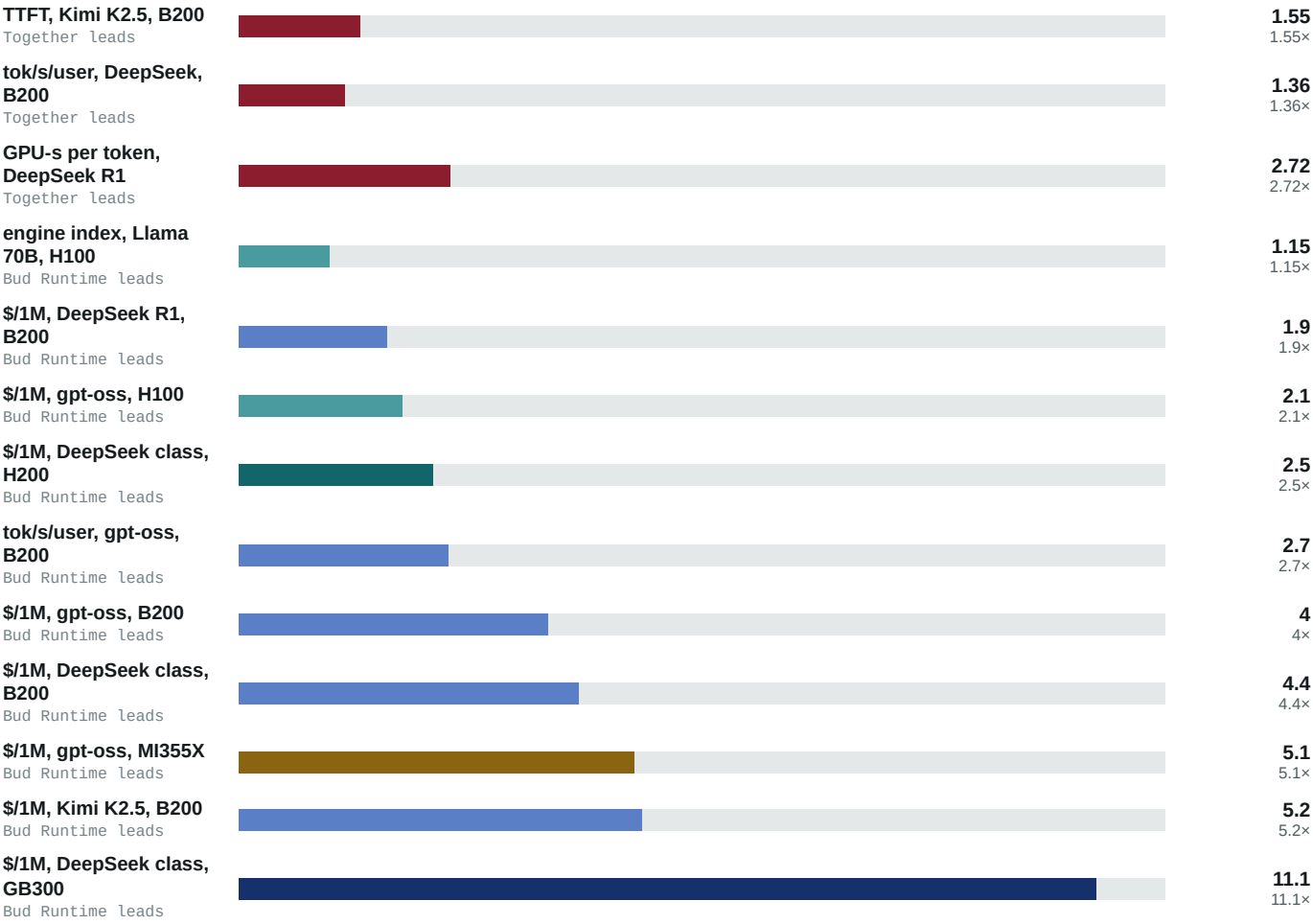
Together's ATLAS reaches 58% of the physical decode ceiling on FP8. Bud Runtime's TensorRT path with MTP sits at 11% of a much higher FP4 ceiling, leaving the most unrealised headroom of any configuration in this document.

Cost per 1M	**Per TB/s**	**Ceiling**	**MBU**
rate × GPUs ÷ (tok/s × 3600) × 10^6	tok/s per GPU ÷ HBM bandwidth	aggregate HBM ÷ bytes per token	measured tok/s ÷ ceiling

Head-to-head against Together AI

Every admissible comparison · advantage multiple

red = Together leads · hardware colour = Bud Runtime leads · sorted



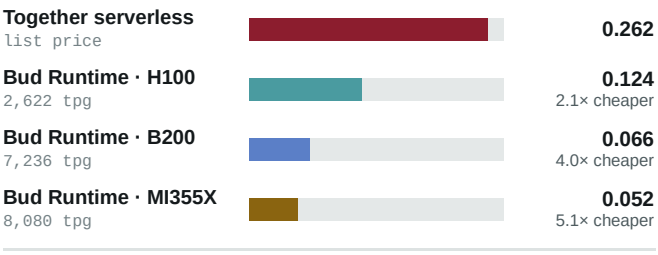
Three cells favour Together, all on B200: first-token latency under agentic load, batch-of-one decode, and GPU-efficiency at batch of one. Every cost cell and every other hardware family favours Bud Runtime, by 1.9× to 11.1×.

Hardware	Model	Regime	Together	Bud Runtime	Metric	Ratio	Leader
B200	Kimi K2.5	R3	0.71 s	1.10 s	TTFT p50	1.55×	Together
B200	Kimi K2.5	R3	$0.240	$0.046	$/1M, same throughput	5.2×	Bud Runtime
B200	DeepSeek	R1	501 tpu · 4 GPU	368 tpu · 8 GPU	tok/s per user	1.36×	Together
B200	DeepSeek	R1	7.98 ms	21.7 ms	GPU-seconds per token	2.72×	Together
B200	DeepSeek	R1	$19.94	$10.45	$/1M	1.9×	Bud Runtime
B200	gpt-oss-120B	R3 vs list	$0.262	$0.066	$/1M	4.0×	Bud Runtime
MI355X	gpt-oss-120B	R2 vs list	$0.262	$0.052	$/1M	5.1×	Bud Runtime
H100	gpt-oss-120B	R2 vs list	$0.262	$0.124	$/1M	2.1×	Bud Runtime
B200	gpt-oss-120B	R1	~187 tpu shared	>500 tpu dedicated	tok/s per user	~2.7×	Bud Runtime
GB300	DeepSeek class	R2 vs list	$1.98	$0.178	$/1M	11.1×	Bud Runtime
B200	DeepSeek class	R2 vs list	$1.98	$0.454	$/1M	4.4×	Bud Runtime
H200	DeepSeek class	R2 vs list	$1.98	$0.803	$/1M	2.5×	Bud Runtime
H100	Llama 70B	index	4.00×	4.59×	vs vLLM 0.5.1	1.15×	Bud Runtime
B200	70B training	T	15,264 tpg	—	tok/s per GPU	—	Together

Cost

gpt-oss-120B · cost per 1M tokens

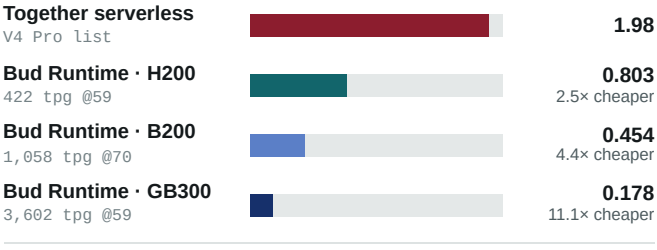
identical MXFP4 open weights · blended 3:1 · Bud Runtime at 100% duty



Same open weights, same MXFP4 precision. Bud Runtime is 2.1× cheaper on H100, 4.0× on B200 and 5.1× on MI355X against Together's list price at full utilisation.

DeepSeek class · cost per 1M tokens

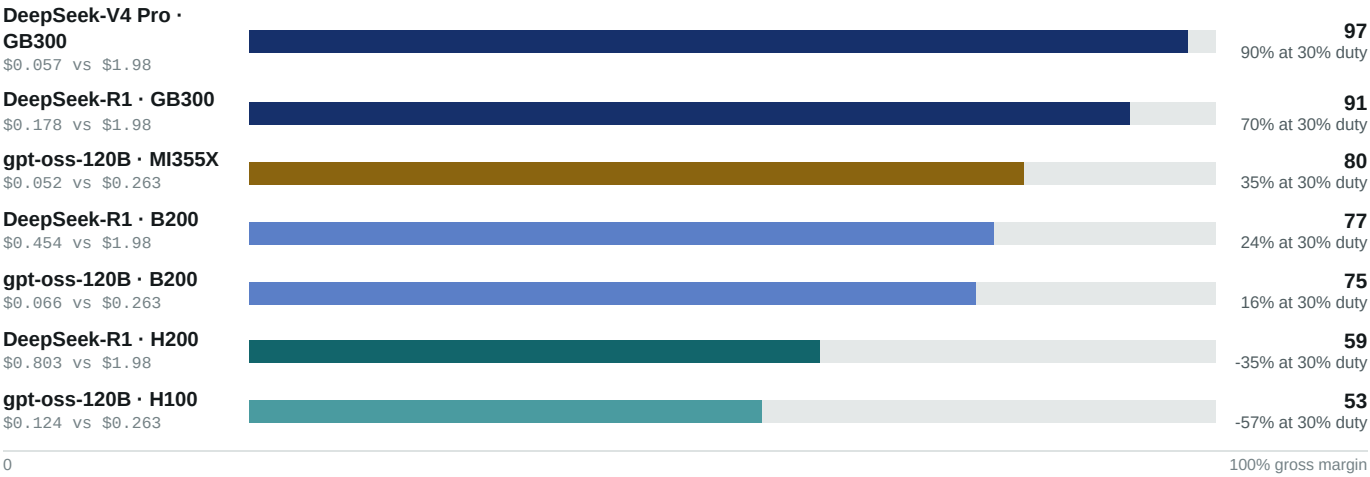
Together V4 Pro list vs Bud Runtime R1-class · blended 3:1



Against Together's DeepSeek V4 Pro list price, Bud Runtime's R1-class deployments cost 2.5× less on H200 and 11.1× less on GB300, the hardware Together does not sell. Models are class-matched, not identical.

Gross margin reselling at Together AI list price

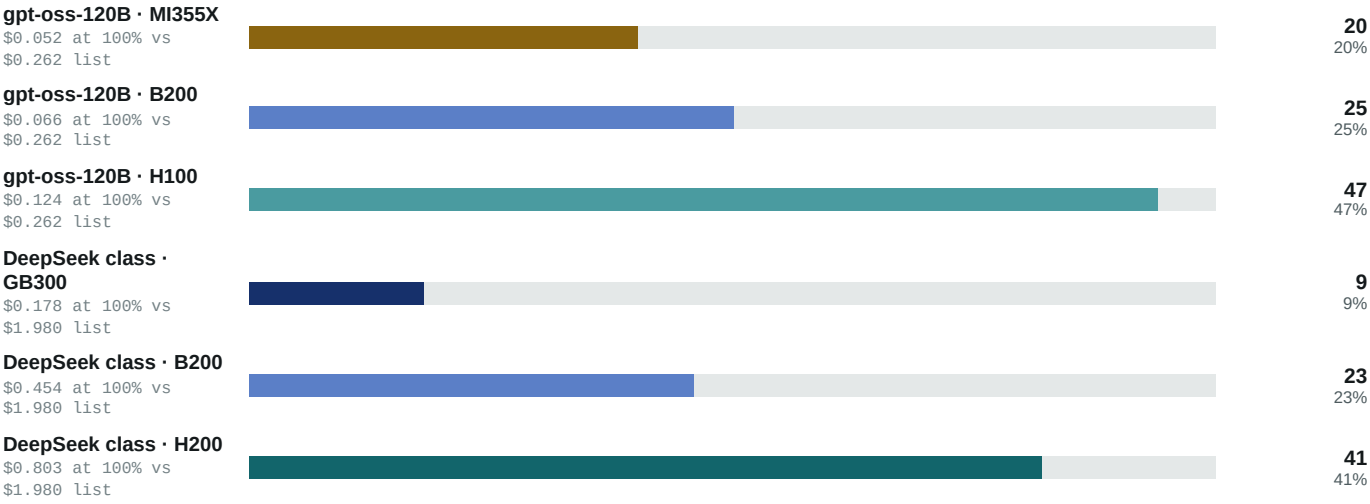
Bud Runtime cost against Together list, blended 3:1 · bar = full utilisation · tag = 30% duty



A provider reselling Bud Runtime capacity at Together's list price earns 53–91% gross margin at full utilisation, and 70–83% on Blackwell Ultra even at 30% duty. Hopper margins turn negative below roughly 45% duty, which is where the SLO scheduler earns its place.

Break-even duty cycle · below this, Together list price is cheaper

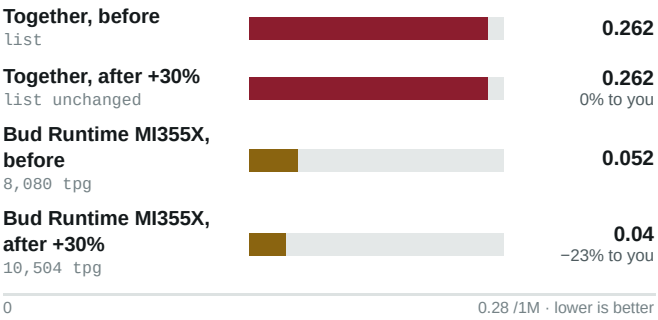
sustained GPU utilisation at which Bud Runtime cost equals Together list



Bud Runtime wins from 9% sustained utilisation on GB300 and 20% on MI355X; the H100 gpt-oss case needs 47%. Bud Runtime's SLO scheduler exists to keep fleet utilisation above these lines by batching across SLO classes.

Whose efficiency gains reach the invoice

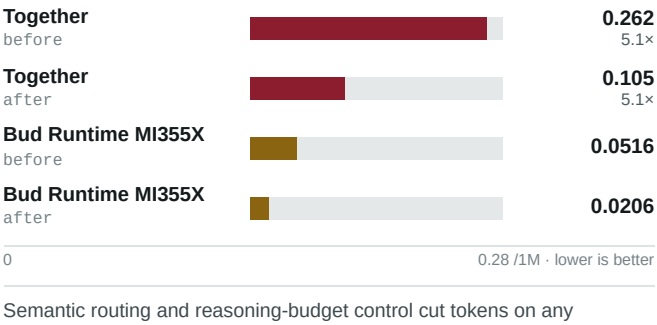
a 30% engine gain applied to both sides · gpt-oss-120B



Every engine improvement Bud Runtime ships lowers the customer's bill directly: a 30% gain is a 23% cost cut. The same gain inside Together lowers Together's margin and leaves list price where it was.

Provider-neutral savings leave the ratio unchanged

semantic routing + reasoning budget, −60% tokens



Semantic routing and reasoning-budget control cut tokens on any provider. Bud Runtime ships both; they lower both bills by 60% and leave Bud Runtime's 5.1× cost advantage exactly where it was.

Optimisation stack

Seventeen optimisation families · who applies what

filled = applied · six on both sides cancel · one proprietary to Together

	Bud Runtime	Together AI
EAGLE-3 / MTP speculation	■	■
Prefill–decode disaggregation	■	■
FP8 / NVFP4 quantisation	■	■
FlashAttention-3 / 4	■	■
ThunderMLA, megakernels	■	■
Prefix / prompt caching	■	■
LMCache + CacheBlend non-prefix KV reuse	■	□
Wide-EP + DeepEP + EPLB / Waterfill	■	□
DSA / sparse-attention backend	■	□
KV tiering DRAM / NVMe	■	□
SpecForge custom drafts	■	□
AIConfigurator config search	■	□
GB300 / B300 deployment	■	□
AMD MI355X deployment	■	□
Adaptive runtime-learning speculator	□	■
Semantic routing	▪	▪
Reasoning-budget control	▪	▪

■ applied □ not applied / not disclosed

▪ customer-side, provider-neutral

Six families sit on both sides and cancel. Bud Runtime applies eight that Together does not disclose or does not sell, two of which are hardware access. Together's single proprietary lead is its runtime-learning speculator.

Bud Runtime configuration matrix · per model

optimisations Bud Runtime enables for each model family

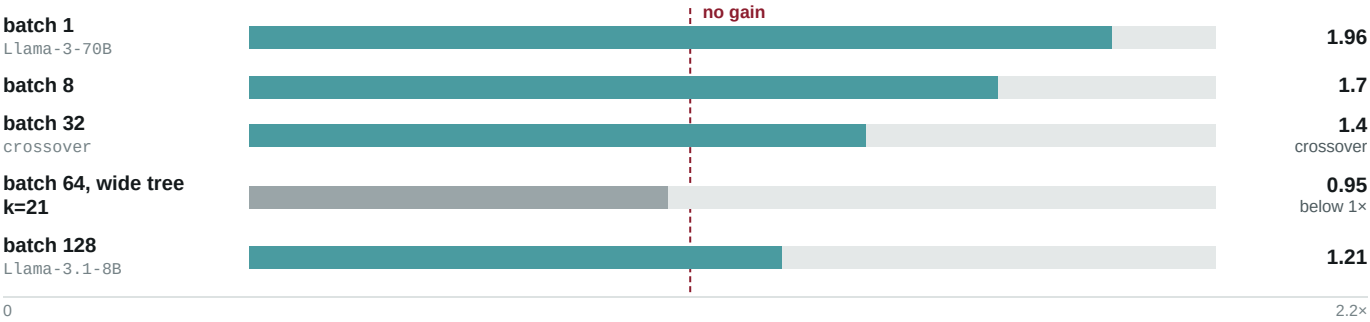
	NVFP4 / FP8	Wide-EP	DSA	PD disagg	MTP / EAGLE-3	LMCache	AITER (AMD)
DeepSeek V3.2 / V4	■	■	■	■	■	■	□
Kimi K2.x / K3	■	■	□	■	■	■	■
GLM-5.x	■	■	■	■	■	■	□
Qwen3.5 / Next	■	■	□	■	■	■	□
MiniMax M3	■	■	■	■	□	■	□
gpt-oss-120B	■	□	□	□	■	■	■
Llama 70B	■	□	□	□	■	■	■

■ enabled □ not applicable

Bud Runtime selects per model family: wide expert parallelism and sparse attention for MoE, EAGLE-3 drafts for dense models, LMCache for every multi-turn workload, AITER on AMD. One orchestrator, one API, best configuration per cell.

Speculative decoding across batch size · why Bud Runtime enables drafts only on low-batch replicas

EAGLE chain speedup on H100 · wide trees fall below 1× at batch 64

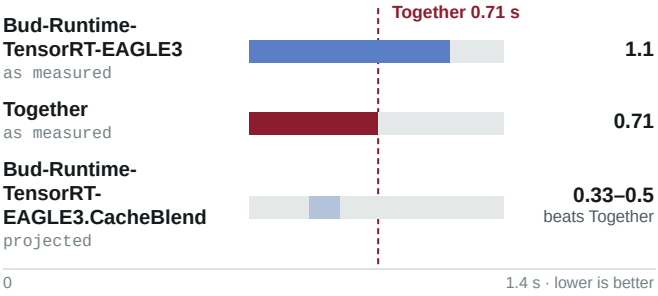


Speculative decoding's 2× at batch 1 falls to 1.2× at batch 128 and below 1× for wide trees at batch 64. Bud Runtime's scheduler enables drafts only on low-batch replicas, so customers keep the gain without paying the loss.

Projections · not measured

Kimi K2.5 · TTFT with non-prefix KV reuse

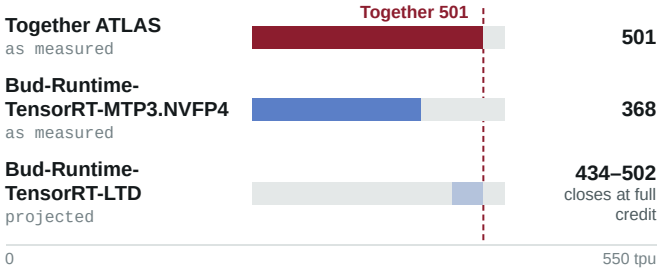
B200 · CacheBlend 2.2–3.3× on reuse-heavy traffic · projected



Together's TTFT benchmark used no cross-request KV reuse on 45k–200k-token multi-turn prompts, the ideal case for it. Bud Runtime's LMCache with CacheBlend path projects to 0.33–0.50 s, ahead of Together's 0.71 s.

DeepSeek · batch of one with adaptive drafts

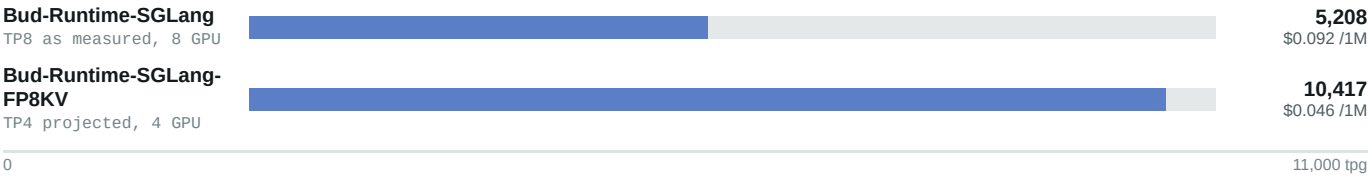
B200 · Learning-to-Draft +36.4% over EAGLE-3 · half to full credit · projected



Bud Runtime's adaptive-draft path projects to 434–502 tok/s per user, matching Together's ATLAS at full credit. The projection transfers a gain measured on a different model family and is marked low confidence accordingly.

Kimi K2.5 · SGLang path with quantised KV cache

B200 · TP8 result was an out-of-memory failure at TP4 · FP8 KV halves cache memory



The SGLang result Together benchmarked was an out-of-memory failure forcing eight GPUs. Bud Runtime's FP8-KV configuration fits TP4, doubling per-GPU throughput and halving cost to $0.046 per million.

Cell	Standing	Technique	Technique gain	Projection	Confidence
B200 · Kimi K2.5 · TTFT	1.10 s vs 0.71 s	LMCache + CacheBlend	2.2–3.3× TTFT	0.33–0.50 s	moderate
B200 · DeepSeek · R1	368 vs 501 tpu	Learning-to-Draft	+36.4% over EAGLE-3	434–502 tpu	low, cross-model
B200 · Kimi K2.5 · SGLang	5,208 tpg at TP8	FP8 / NVFP4 KV	~2× KV memory	10,417 tpg at TP4	high, merged flag

Hardware backend coverage

Bud Runtime is hardware-abstracted: the orchestrator selects engine, precision, parallelism and speculation per target, and the engine layer beneath it reaches the silicon below. **This page states backend availability, not measured performance.** The six families carrying measured Bud Runtime figures are set out in the body of this document; every other row here is a supported execution path, and a deployment on one begins with a Bud Simulator configuration run and a benchmark, not with a claim. Together AI publishes performance on two families and sells on NVIDIA alone.

Hardware backend support matrix

`silicon platforms reachable through the engine layer Bud Runtime composes`

Platform family	Silicon and accelerators
NVIDIA CUDA	Volta V100 · Turing T4, RTX 20xx · Ampere A100, RTX 30xx · Ada L4, L40S, RTX 40xx · Hopper H100, H200 · Blackwell B100, B200, GB200, B300, RTX 5090
AMD ROCm	Instinct MI200, MI300A/X, MI355X · RDNA3, gfx950
Google TPU	TPU v4, v5e, v5p, v6e Trillium, v7
Huawei Ascend NPU	Ascend 910, 910B, 910C, 310P
Intel GPU (XPU)	Data Center GPU Max (Ponte Vecchio), Flex, Arc discrete and integrated
Intel Gaudi (HPU)	Gaudi 2, Gaudi 3
AWS Neuron	Inferentia1, Inferentia2, Trainium1, Trainium2
Intel CPU	Xeon Scalable, Skylake through Granite Rapids
AMD CPU	EPYC Zen 2, Zen 3, Zen 4, Zen 5
ARM CPU	AArch64 · AWS Graviton 2/3/4, Ampere Altra, Neoverse
Apple Silicon	M1, M2, M3, M4 and Pro / Max / Ultra
Rebellions NPU	ATOM, REBEL
IBM Spyre AIU	Spyre AI accelerator
Baidu Kunlun XPU	Kunlun3 P800
MetaX MACA GPU	MetaX MACA GPUs
Tenstorrent NPU	Wormhole, Grayskull
Cambricon MLU	Cambricon MLU accelerators
IBM Z mainframe	S390X processor architecture

6 families measured GB300/B300, B200, H200, H100, MI355X, MI325X	**18 platform families** reachable through the engine layer	**Non-NVIDIA routes** AMD, Ascend, TPU, Gaudi, Neuron, CPU	**1 orchestrator** one API and one artefact across all of them

Modality coverage

Inference economics are argued on text throughput, but an enterprise or government estate does not consist of text. It consists of recorded calls, scanned documents, screen workflows, sensor histories and forecasts. **Bud Runtime serves eleven modality classes on one orchestrator, one API and one fleet**; Together AI serves four of them and offers no route to the rest. Where a second modality means a second platform, the cost comparison in this document understates the difference — a provider standing up separate voice, document and forecasting stacks pays for them in integration, idle capacity and operational surface, not in cost per million tokens.

Modality coverage · Bud Runtime against Together AI

`model classes served on the platform · Together AI catalogue and documented endpoints, September 2026`

Modality class	Representative model families	Bud Runtime	Together AI
Text and reasoning	frontier open-weight LLMs and MoE reasoning models	served	served
Speech to text	transcription and streaming ASR	served	served · Whisper Large v3, batch and streaming
Text to speech	neural TTS and voice cloning	served	served · Sonic 3, Orpheus 3B, Kokoro
Speech to speech	full-duplex audio-in, audio-out voice models	served · **Bud WaaV**	**not offered** · voice must be assembled from separate transcription and synthesis calls
Embeddings and retrieval	encoder models, re-rankers, late-interaction retrievers	served	served · embeddings and rerank endpoints
Vision encoders and segmentation	self-supervised vision backbones and promptable segmentation	served	**not offered**
Document understanding	OCR and document-layout models	served	**not offered as a class** · vision-language models only
Actions and GUI control	screen grounding and computer-use action models	served	**not offered**
World models	joint-embedding predictive architectures	served	**not offered**
Weather and earth systems	diffusion ensemble forecast models	served	**not offered**
Time series	foundation forecasting models	served	**not offered**

Image and video generation are the reverse case: Together AI publishes image and video model endpoints, and they sit outside the modality set stated here. Every non-text class Bud Runtime serves reaches the same orchestrator, the same hardware abstraction and the same SLO scheduler as the text path, so a voice model and a forecasting model are scheduled on the same fleet rather than on a second one.

11 modality classes served by Bud Runtime on one runtime	**4 served by both** text, transcription, synthesis, embeddings	**7 Bud Runtime only** voice-native, vision encoders, documents, actions, world, weather, time series	**1 API, 1 fleet** no second platform per modality