GPU SMITH
COMPUTE ENGINEERING

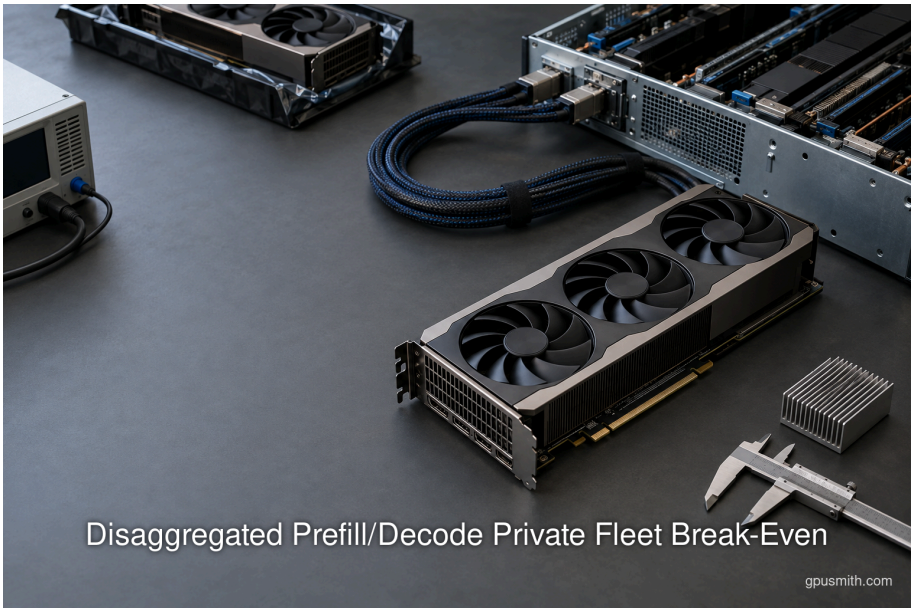
GPU SMITH / RESEARCH & PRACTICAL GUIDANCE

Disaggregated Prefill/Decode Private Fleet Break-Even

> Plan and build private AI compute with GPU Smith. We connect workload requirements with GPU hardware, networking, deployment and operating decisions so your infrastructure fits the work it must perform.

Published by GPU Smith · 2026-09-22 · disaggregated prefill decode · prefill decode disaggregation · llm inference · gpu fleet sizing · vllm · nvidia dynamo · tensorrt-llm · ray serve · kv cache

Original article: https://gpusmith.com/articles/en/disaggregated-prefill-decode-break-even



In this article

Executive Summary

Disaggregated prefill/decode is not automatically cheaper than colocated inference. It is a capacity-allocation choice: reserve one GPU pool to process prompts, reserve another to generate tokens, and pay to move the key-value (KV) cache between them. The decision should therefore be made on **qualified goodput**, the rate of requests that satisfy both time to first token (TTFT) and inter-token latency (ITL) objectives, rather than raw tokens per second. DistServe similarly defines goodput around the maximum request rate that meets a service-level-objective (SLO) attainment target (arxiv.org). As of **September 22, 2026**, NVIDIA Dynamo **v1.5.0** pairs with vLLM **0.28.0**, SGLang **0.5.18**, and TensorRT-LLM **1.3.0rc25** (docs.nvidia.com). Those versions are the dated software baseline in this report, not a promise that later releases behave identically.

Three conditions justify a controlled disaggregation test: long or bursty prompts cause decode streams to miss ITL, prefill and decode need materially different parallelism or hardware, and the measured SLO gain pays for KV transfer plus a second minimum pool. Three conditions favor aggregation: low concurrency, short prompts, or insufficient transfer bandwidth. NVIDIA documents that an aggregated worker often avoids transfer and pool-fragmentation overhead at low concurrency (docs.nvidia.com); vLLM cautions that disaggregated prefill does not itself improve throughput (docs.vllm.ai). **Conditional routing** is the middle path: keep short or cache-rich requests on decode workers and disaggregate only requests predicted to benefit. Dynamo explicitly describes this as a hybrid of aggregated and disaggregated routing (docs.nvidia.com).

The proposed private-fleet worksheet uses one declared enterprise retrieval-augmented generation (RAG) and chat trace, identical weights and traffic for all designs, and measured phase throughput rather than vendor peaks. Required pool GPUs equal `CEILING(required phase token throughput / measured phase token throughput per GPU)`. Monthly GPU-hours equal `pool GPUs × scheduled hours`. Qualified goodput equals requests meeting both SLOs divided by wall-clock seconds. Cost per million accepted output tokens equals total period infrastructure cost divided by accepted output tokens, multiplied by **1,000,000**. The fixed xPyD design wins only when its lower cost per accepted token, or greater accepted demand at the same cost, exceeds the penalties from the extra pool, idle capacity, KV transfer, and HBM removed from decode.

The baseline uses **H200 SXM** accelerators and deliberately leaves acquisition cost as a dated quote input. For energy sensitivity, the June 2026 US industrial average was **9.17 cents/kWh** (www.eia.gov). A production commitment requires measured crossover curves, raw summaries, and rollback gates, not adoption of a vendor diagram or a research-paper headline. Ray Serve's current implementation also requires its default vLLM v1 engine, which illustrates why engine generation belongs in the test contract (docs.ray.io).

Introduction and Background

Large language model inference has two operationally different phases. **Prefill** processes the input tokens and creates the KV cache. **Decode** repeatedly reads model weights and growing state while producing output tokens. Splitwise characterizes this distinction as compute-intensive prompt work followed by memory-intensive token generation (www.microsoft.com). Colocation shares GPUs and schedulers between them. Disaggregation assigns them to independent GPU pools, then transfers the KV state.

That separation can prevent a long prompt from interrupting active streams, but it introduces four costs that architecture diagrams often leave implicit:

- **Minimum-pool cost:** a deployable xPyD layout needs at least one prefill replica and one decode replica, each rounded to its model-parallel GPU count.
- **Transfer cost:** KV bytes consume fabric bandwidth and add time between phase completion and useful decode.
- **Mix risk:** a fixed pool can be idle when the input-to-output ratio shifts, even while the other pool queues.
- **HBM opportunity cost:** every GPU reserved for prefill removes high-bandwidth memory capacity that could admit more active decode sequences.

The last effect is material enough that NVIDIA warns additional prefill replicas can reduce total capacity when decode KV memory is limiting (docs.nvidia.com). Small installations also have fewer allocation choices because replica granularity consumes a larger fraction of the fleet.

This report treats the decision as a reproducible capacity-planning exercise for enterprise vLLM, TensorRT-LLM, Ray Serve, or Dynamo clusters. In practical terms, it connects **prefill decode disaggregation cost**, **prefill vs decode GPU requirements**, and **LLM inference GPU fleet sizing** in one disaggregated inference architecture model. It also answers when separate GPU pools for LLM inference reach a disaggregated serving break even point within private LLM infrastructure economics. GPU Smith is an **adjacent independent engineering advisor**, not a serving engine or GPU manufacturer. Its stated method derives topology, node count, and serving stack from workload models and throughput targets (gpusmith.com); that is the appropriate role here. The firm also states that it has no reseller quota or cloud of its own (gpusmith.com), which is relevant to keeping the worksheet vendor-neutral. The output is a test contract and break-even worksheet, not a universal recommendation.

Key Changes

From one shared queue to two capacity domains

An aggregated replica lets its scheduler interleave prefill and decode. A disaggregated deployment creates two queueing systems and a handoff. The gain is isolation and independent scaling. The price is integer rounding, transfer, and more failure and observability boundaries.

The architectural choice should be tested when all three of these conditions are present:

- **Observable interference:** p95 or p99 ITL deteriorates when long prompts arrive.
- **Separately tunable phases:** prefill and decode benefit from different tensor parallelism, batching, or GPU types.
- **Economic headroom:** the accepted-token gain can exceed the cost of the smallest additional pool.

Aggregation is the leading candidate when any of these dominates:

- **Low concurrency:** active decode streams rarely overlap expensive prefill.
- **Short, homogeneous prompts:** transfer and routing become a large share of service time.

- **Small fleet:** replica granularity makes a second pool disproportionately expensive.

AWS presents the same distinction in implementation terms: its documented example keeps a request local below a threshold and sends a **6,750-token** request through disaggregation at **4,096 tokens** (docs.aws.amazon.com). That threshold is implementation guidance, not a physical constant. It must be fitted to the target trace, cache hit rate, fabric, and SLO.

From nominal throughput to qualified goodput

Raw output tokens per second can rise while user-visible service worsens. The declared p95 TTFT and p95 per-request mean-ITL limits remain the service SLO and must be reported as SLO attainment. For accepted-token economics, however, this worksheet intentionally uses a stricter per-request acceptance rule: a request counts only when both its own TTFT and its own mean ITL meet those limits. Apply that stricter rule consistently to qualified goodput, accepted output tokens, cost per million accepted output tokens, and crossover comparisons. vLLM defines mean ITL as the average interval between successive output tokens during decode (docs.vllm.ai). A test must state whether the SLO uses per-request mean ITL, worst-gap ITL, or a distribution across token gaps. These are not interchangeable.

Use the following primary metric:

```
qualified_goodput_rps = count(request_ttft <= TTFT_SLO AND request_itl <= ITL_SLO) / test_wall_seconds
```

Also report **SLO attainment**, accepted output tokens per second, p50/p95/p99 TTFT, p50/p95/p99 ITL, queue time, and error rate. vLLM exposes an inter-token-latency histogram between consecutive streamed outputs (docs.vllm.ai), as well as per-request prefill and decode histograms (docs.vllm.ai) (docs.vllm.ai).

From fixed disaggregation to conditional routing

Fixed xPyD sends every request through a prefill worker, then a decode worker. **Conditional disaggregation** decides whether local prefill on a decode worker is cheaper than remote prefill plus transfer. Dynamo requires visibility into decode-side KV events for that estimate (docs.nvidia.com).

Routing inputs should include:

- **Estimated uncached input tokens**, not only total prompt length.
- **Decode occupancy**, including active sequences and KV free space.
- **Predicted transfer time**, based on recent effective bandwidth rather than link rate.
- **Output-length estimate**, because a long decode may justify protecting ITL.
- **Pool queue delay**, so an idle decode worker can absorb prefill during a prefill burst.

Recent research reinforces the hybrid case. Splitwise found that a mixed prompt-and-token pool becomes useful at higher loads (arxiv.org). Kairos conditionally executes prefill on decode nodes only when predicted time-between-token latency remains safe, reporting up to **81%** lower p95 TTFT and **79%** higher SLO

attainment in its stated DeepSeek-V2-Lite experiments (arxiv.org). Those are research results for specific traces, not private-fleet expectations.

Workload Contract and Baseline Measurement

Declared planning trace (Hypothetical Example)

The worksheet needs one immutable workload contract. The following is a **hypothetical enterprise RAG/chat trace**, designed to make every assumption visible. It is not a benchmark observation.

Table 1 defines the workload and dated test configuration that every candidate architecture must share.

Input	Declared value	Measurement rule
Model and precision	One 70B-class instruction model, FP8 weights	Same checkpoint hash, tokenizer, quantization artifact, maximum context, and sampling parameters for every run
Software snapshot	Report snapshot above, with CUDA and driver versions recorded at test time	Store container digests and full launch commands
Hardware	Four eight-GPU H200 SXM nodes, identical CPU, RAM, NIC, and storage	Record serializable node inventory; H200 is specified at 141 GB HBM3e per GPU (docs.nvidia.com)
Arrival profile	1.2 requests/s steady, 2.4 requests/s for 15 minutes in each hour	Replay the original timestamps; do not replace arrivals with closed-loop concurrency
Input tokens	p50 2,000; p95 8,000; p99 16,000	Preserve full empirical histogram and prompt-prefix reuse
Output tokens	p50 250; p95 900; p99 1,500	Preserve stop reasons and maximum-token caps
Long-prompt share	30% above 4,096 uncached tokens	Sweep 10% through 70% in sensitivity testing
SLO	p95 TTFT <= 2.0 s and p95 per-request mean ITL <= 50 ms	A request is accepted only if both limits are met
Scheduling period	730 hours/month	Separate scheduled from actually available hours
Electricity	$0.0917/kWh, then sweep local tariff	June 2026 US industrial average was 9.17 cents/kWh (www.eia.gov)
Acquisition input	Buyer quote `Q`, dated, fully configured and installed	No universal MSRP assumed; amortize over the organization's accounting life

This contract intentionally combines long-context bursts with substantial output lengths. It gives prefill and decode a chance to become different bottlenecks. The result still applies only to this histogram, software snapshot, topology, and policy. Mooncake's production-derived one-hour trace, for comparison, averaged **7,590 input** and **182 output tokens** (arxiv.org), illustrating how sharply real phase ratios can differ.

Aggregated baseline protocol

The baseline is not the fastest vendor example. It is the best stable aggregated configuration found under the same SLO and hardware budget.

1. **Freeze artifacts:** record image digests, model hash, tokenizer hash, driver, firmware, engine flags, and router configuration.
2. **Warm identically:** use the same warm-up duration and prefix-cache state before every measured window.
3. **Sweep load:** replay the trace at 0.5, 0.75, 1.0, 1.25, and 1.5 times target arrival rate.
4. **Tune fairly:** sweep tensor parallelism, batch limits, chunked-prefill size, scheduler policy, and KV allocation within a fixed tuning budget.
5. **Repeat:** run at least three windows per operating point and retain request-level records.
6. **Select by goodput:** choose the configuration with maximum qualified goodput, not maximum offered throughput.
7. **Preserve failures:** count timeouts, rejected requests, and malformed responses in the denominator.

vLLM defines its TTFT field from scheduling until the first generated output token (docs.vllm.ai). The load generator's end-to-end TTFT will additionally include ingress, routing, and network time. Store both; the user-facing metric is authoritative for acceptance.

xPyD measurement protocol

Start with the same total GPU budget and enumerate feasible prefill/decode allocations. A model-parallel replica size of four GPUs, for example, permits **1P7D**, **2P6D**, and other combinations on a 32-GPU fleet. It does not permit fractional replicas. This integer constraint is the extra-minimum-pool penalty.

For every layout:

- **Measure phase capacity:** accepted input tokens/s per prefill GPU and accepted output tokens/s per decode GPU.
- **Capture KV bytes:** use engine telemetry per request, not a theoretical peak. vLLM exposes `nixl_bytes_transferred` for each KV transfer (docs.vllm.ai).
- **Capture transfer duration:** vLLM also exposes NIXL transfer-duration histograms (docs.vllm.ai).
- **Verify transport:** HyperPod's implementation uses EFA with GPU-Direct RDMA (docs.aws.amazon.com); other environments must prove their actual data path.
- **Test failure handling:** restart a prefiller, decoder, router, and NIC path during controlled runs.
- **Hold routing constant:** compare fixed disaggregation first, then introduce conditional routing as a separate candidate.

Backend semantics matter. Dynamo documents vLLM prefill as synchronous in its handoff flow (docs.nvidia.com), while the underlying vLLM NixlConnector supports fully asynchronous send and receive (docs.vllm.ai). TensorRT-LLM overlaps transfer for one request with computation for independent requests (nvidia.github.io), but its Python transceiver currently moves the whole prompt only after prefill completes (nvidia.github.io). The benchmark must describe both request-level synchronization and lower-level transfer overlap.

Implementation Considerations and Process Changes

Fabric and placement

KV transfer time is approximately `measured_bytes / measured_effective_bandwidth + fixed_handoff_latency`. Link speed is only a ceiling. NUMA placement, PCIe topology, GPU Direct, congestion, and serialization determine effective bandwidth.

Useful deployment checks are:

- **Same-node path:** validate NVLink or equivalent peer access and record topology.
- **Cross-node path:** validate RDMA, GPU memory registration, MTU, congestion control, and route symmetry.
- **Bandwidth headroom:** compare p95 KV bytes/s with sustained measured bandwidth, not adapter label.
- **Failure domain:** avoid a topology in which one fabric fault removes both phase pools.
- **Telemetry clocking:** synchronize nodes so queue, prefill, transfer, and decode spans can be joined.

ConnectX-7 adapters are offered with **400 Gb/s or 200 Gb/s** connectivity (networking-docs.nvidia.com), but an adapter rating is not request-level throughput. A second recorded adapter check should confirm that the installed port is the intended 400 Gb/s or 200 Gb/s variant (networking-docs.nvidia.com). DistServe's OPT-66B example estimated **11.3 GB/s**, or **90 Gb/s**, to hide transfer at its assumed request rate (arxiv.org). That number belongs to its stated model, prompt length, and traffic, not this planning trace.

HBM and decode opportunity cost

Decode capacity is constrained by weights, runtime workspace, and live KV state. Removing one H200 from decode removes **141 GB** of gross HBM and **4.8 TB/s** of specified bandwidth from the decode pool (docs.nvidia.com). The correct cost is not merely that GPU's amortization. It includes accepted output tokens that the removed HBM would have supported during the decode peak.

Record:

- **Weights and workspace bytes per rank.**
- **KV bytes per active request and per context bucket.**
- **Maximum admitted sequences at the SLO-safe operating point.**

- **Eviction, recomputation, and prefix-cache hit rates.**
- **Rejected output tokens caused by decode admission limits.**

Define monthly decode opportunity cost as `prefill_reserved_GPUs × measured_decode_goodput_per_GPU × scheduled_seconds × value_per_accepted_request`, or keep it in physical units as foregone accepted output tokens. Do not add both forms to total cost, which would double count the same loss.

Power, node cost, and availability

Use metered node power if possible. A Cisco eight-GPU H100/H200 system specification gives approximate system power of **12.5 kW** (www.cisco.com), which is a planning reference, not a substitute for the target server's meter. The same 12.5 kW specification is useful as a sanity check against an incorrectly scaled meter export (www.cisco.com). NVIDIA Triton exposes cumulative per-GPU energy since server start in joules (docs.nvidia.com). One kilowatt-hour means one kilowatt sustained for one hour (www.energy.gov). Convert energy consistently, then apply facility overhead using power usage effectiveness (PUE), defined as total data-center energy divided by IT energy (www.energystar.gov).

Cost inputs should include:

- **Amortized compute:** dated installed node quote divided by useful-life months.
- **Fabric allocation:** switches, adapters, optics, cabling, support, and spares allocated to the serving fleet.
- **Power:** measured kWh multiplied by the contracted tariff and PUE where appropriate.
- **Software and operations:** licenses, orchestration, monitoring, and on-call labor if they differ by design.
- **Availability:** subtract maintenance, failures, and deployment windows from scheduled hours.

Vendor price scenarios must carry dates. Lenovo, for example, labels its June 2026 figures as usual customer sale prices as of **June 15, 2026** (lenovopress.lenovo.com). That makes them dated scenarios, not universal MSRP. The worksheet therefore requires the enterprise's own quote.

Data Analysis and Evidence

Copyable break-even worksheet

Table 2 is a spreadsheet-ready model. Inputs marked "measured" must come from the trace replay. Calculated scenarios must remain labeled separately from observations.

Row	Variable	Spreadsheet formula or source
1	Required prefill tokens/s	`accepted_request_rate × mean_uncached_input_tokens × burst_factor`
2	Required decode tokens/s	`accepted_request_rate × mean_output_tokens × burst_factor`

Row	Variable	Spreadsheet formula or source
3	Measured prefill tokens/s/GPU	Observed at the selected TTFT and ITL SLO
4	Measured decode tokens/s/GPU	Observed at the selected TTFT and ITL SLO
5	Prefill pool GPUs	`CEILING(row1 / row3, replica_GPU_count)`
6	Decode pool GPUs	`CEILING(row2 / row4, replica_GPU_count)`
7	Aggregated GPUs	`CEILING(target_qualified_goodput / measured_aggregate_goodput_per_GPU, replica_GPU_count)`
8	KV bytes/request	Mean and percentile from engine telemetry; vLLM provides a per-transfer bytes histogram (docs.vllm.ai)
9	KV transfer seconds/request	`measured_transfer_end - measured_transfer_start`; AWS exposes an explicit KV transfer-time metric (docs.aws.amazon.com)
10	Monthly GPU-hours	`(prefill_pool_GPUs + decode_pool_GPUs) × scheduled_hours`
11	Idle pool-hours	`SUM(MAX(0, provisioned_phase_GPU_equivalents - demanded_phase_GPU_equivalents) × interval_hours)`
12	Qualified goodput	`requests_meeting_both_TTFT_and_ITL / wall_clock_seconds`
13	Accepted output tokens	`SUM(output_tokens WHERE request_meets_both_SLOs)`
14	Monthly amortized cost	`installed_capex / useful_life_months + fabric_allocation + software + operations`; record the quote date because vendor scenarios can be explicitly date-bound (lenovopress.lenovo.com)
15	Monthly energy cost	`measured_IT_kWh × applicable_PUE × tariff_per_kWh`; PUE is total data-center energy or power divided by IT energy or power (www.energystar.gov)
16	Total period cost	`row14 + row15 + variable_operating_costs`
17	Cost/million accepted output tokens	`row16 / row13 × 1000000`
18	Break-even delta	`aggregate_cost_per_million - candidate_cost_per_million`

Integer rounding in rows 5 through 7 is central. If the model requires four GPUs per replica, demand for 4.1 GPU-equivalents becomes eight GPUs, not five. The unused fraction appears in idle pool-hours and must not disappear inside an average utilization figure. GPU Smith states that its engagements are delivered against written acceptance criteria and an as-built documentation set (gpusmith.com); the worksheet can provide the quantitative core of those criteria without treating the consultancy as a serving option.

Crossover chart data series (Hypothetical Example)

The following is a **hypothetical chart template**, not benchmark data. Paste the long-prompt shares into a spreadsheet, recompute phase demand and accepted-token cost from measured cells, then create a line chart with long-prompt share on the x-axis and cost per million accepted output tokens on the y-axis.

Long-prompt share	Aggregate series	Fixed xPyD series	Conditional series
10%	`=AggregateCost/AcceptedTokens_A10*1000000`	`=FixedCost/AcceptedTokens_F10*1000000`	`=ConditionalCost/AcceptedTokens_C10*1000000`
20%	`=AggregateCost/AcceptedTokens_A20*1000000`	`=FixedCost/AcceptedTokens_F20*1000000`	`=ConditionalCost/AcceptedTokens_C20*1000000`
30%	`=AggregateCost/AcceptedTokens_A30*1000000`	`=FixedCost/AcceptedTokens_F30*1000000`	`=ConditionalCost/AcceptedTokens_C30*1000000`
40%	`=AggregateCost/AcceptedTokens_A40*1000000`	`=FixedCost/AcceptedTokens_F40*1000000`	`=ConditionalCost/AcceptedTokens_C40*1000000`
50%	`=AggregateCost/AcceptedTokens_A50*1000000`	`=FixedCost/AcceptedTokens_F50*1000000`	`=ConditionalCost/AcceptedTokens_C50*1000000`
60%	`=AggregateCost/AcceptedTokens_A60*1000000`	`=FixedCost/AcceptedTokens_F60*1000000`	`=ConditionalCost/AcceptedTokens_C60*1000000`
70%	`=AggregateCost/AcceptedTokens_A70*1000000`	`=FixedCost/AcceptedTokens_F70*1000000`	`=ConditionalCost/AcceptedTokens_C70*1000000`

The crossover is the first workload bucket in which a candidate both meets the SLO and has lower cost per accepted token than the tuned aggregate. Conditional routing can cross earlier because it avoids paying transfer for requests that do not benefit, but that proposition must be measured. AWS's published router example uses a 4,096-token threshold, yet its own logged 6,750-token decision only demonstrates that implementation, not a universal crossover (docs.aws.amazon.com). Fixed xPyD can also cross back above aggregation when longer outputs turn decode HBM into the dominant constraint.

Sensitivity analysis

Sweep one factor at a time around the complete baseline, then test interactions that are operationally plausible:

- **Long-prompt share:** 10%, 30%, 50%, and 70%.
- **Mean output length:** 0.5, 1.0, 1.5, and 2.0 times the trace.
- **Effective KV bandwidth:** 25%, 50%, 75%, and 100% of the measured uncongested rate.
- **Burst factor:** 1.0, 1.5, 2.0, and 3.0 times steady arrivals.
- **Prefix-cache reuse:** original trace, cache disabled, and controlled high-reuse case.
- **Pool ratio:** every feasible xPyD combination at fixed total GPUs.

- **Routing threshold:** uncached-token buckets rather than one inherited 4,096-token rule.

Published benchmarks bound plausibility but cannot fill the worksheet. DistServe's 13B, 512-input, 64-output example reported about **1.6 requests/s** for a colocated A100 80GB system under its stated 90% attainment condition (arxiv.org); its theoretical two-prefill, one-decode allocation reached **10 requests/s**, or **3.3 requests/s/GPU** (arxiv.org). Splitwise reported up to **1.4 times** higher throughput at **20%** lower cost in its simulated comparison (www.microsoft.com). Mooncake reported **20%** and **40%** SLO-compliant throughput improvements on two named workloads with a three-prefill, one-decode setup (arxiv.org). Each result is configuration-specific.

Negative or conditional findings matter equally. TetriInfer reports that its design is not ideal when workloads are heavy in both prefill and decode (arxiv.org). DéjàVu's analytic model finds no disaggregation benefit when prompt KV streaming makes prefill at least twice as costly as the no-streaming case (arxiv.org). These findings support testing the full input/output matrix rather than assuming more long prompts always favor separation.

Acceptance Gates and Rollback Conditions

A production migration should pass all gates over both steady and burst windows. The **SLO goodput gate** passes when the candidate is at least 10% above the tuned aggregate at the same GPU count, or meets demand with fewer total GPU-hours; two consecutive windows below aggregate goodput trigger rollback. The **economics gate** requires at least 8% lower cost per million accepted output tokens after amortization, energy, fabric, and operations; realized savings below 3% trigger review. The **TTFT and ITL gates** require p95 and p99 compliance in every prompt and output bucket; a repeatable critical-bucket regression triggers rollback.

The **KV-transfer gate** requires p99 handoff time within budget and no sustained fabric queue. The **pool-balance gate** requires burst headroom without one pool queueing while the other remains materially idle. The **resilience gate** requires correct recovery from router and worker restarts without lost, duplicated, or malformed responses. The **operability gate** requires dashboards to join queue, prefill, transfer, decode, output, and cost records by request ID. If operators cannot attribute an SLO miss to a phase, the design is not ready for production.

The exact percentage margins in this table are planning policy for the hypothetical example, not industry facts. An enterprise should replace them with its risk tolerance. A narrow apparent advantage is not durable enough when software upgrades, traffic mix, and power prices move.

Rollback must be designed before rollout:

- **Keep aggregated manifests:** preserve the last known-good images and launch configuration.
- **Use progressive traffic:** 1%, 5%, 20%, 50%, then 100%, with automatic SLO evaluation.
- **Retain trace compatibility:** the same request schema and IDs must work on both paths.
- **Drain safely:** stop new routing to a pool, finish active decodes, then release workers.
- **Snapshot evidence:** store raw request summaries and configuration before every change.
- **Revalidate after upgrade:** rerun the crossover because backend transfer behavior can change.

NVIDIA describes Dynamo KV transfer as non-blocking so forward passes for other requests can continue (docs.nvidia.com). TensorRT-LLM likewise documents overlap between KV transmission and computation for independent requests (nvidia.github.io). These features can reduce visible handoff cost, but they do not eliminate bandwidth consumption, pool imbalance, or recovery requirements.

Implications and Future Directions

The practical conclusion is a three-way policy rather than a binary architecture choice.

- **Remain aggregated** when the tuned colocated engine meets both SLOs, long prompts are rare, or the smallest second pool raises cost per accepted token.
- **Use fixed xPyD** when sustained phase asymmetry is stable, both pools remain well utilized, and transfer is predictably hidden.
- **Use conditional disaggregation** when the workload mixes short and long prompts, cache reuse varies, or bursts cause the limiting phase to change.

Ray Serve's current feature requires the vLLM v1 engine (docs.ray.io). AWS's documented HyperPod images package LMCache **0.4.3**, vLLM **0.19.0**, and NIXL **1.0.0** (docs.aws.amazon.com). Those stacks differ from the report's Dynamo snapshot, demonstrating why version and backend must be explicit rather than collapsed into "vLLM disaggregation."

Future capacity planners should expect routing to become more load-aware and cache-aware. In Kairos's reported two-prefill, two-decode A100 experiments, prefill execution accounted for only **2% to 23%** of p95 TTFT, with queueing and transfer making up the balance (arxiv.org). That result suggests another prefill GPU can be the wrong remedy for high TTFT. The next optimization may instead be admission control, deflection, cache locality, or a better transfer path.

Heterogeneous fleets also deserve testing. H100 SXM is specified up to **700 W** configurable thermal design power (www.nvidia.com), while B200 can be configured up to **1 kW per GPU** (docs.nvidia.com). Different memory capacity, bandwidth, power, and acquisition cost can shift the phase crossover. The worksheet is intentionally hardware-neutral: replace every rate and cost with target-fleet measurements.

Finally, publish enough evidence to reproduce the decision:

- **Trace schema:** arrivals, token counts, prefix identity, stop reason, and SLO outcome.
- **Commands and versions:** container digests, engine flags, router thresholds, and firmware.
- **Raw summary:** per-request phase timings, KV bytes, errors, energy, and availability.
- **Scenario labels:** clearly distinguish observations, calculations, and hypothetical sensitivity cases.
- **Refresh trigger:** rerun after an engine, driver, model, topology, or material traffic change.

Conclusion

Disaggregated prefill/decode breaks even only when its improvement in **SLO-qualified goodput** is worth more than its additional minimum pool, KV-transfer overhead, idle capacity, and lost decode HBM. The decision cannot be read from a vendor architecture diagram or borrowed from a paper’s benchmark. It must be measured on the target model, trace, engine version, fabric, and percentile objectives.

For the declared private H200 planning case, the disciplined sequence is straightforward: optimize the aggregated baseline, enumerate integer xPyD layouts, measure phase capacity and KV transfer, price scheduled and idle GPU-hours, then plot cost per million accepted output tokens across long-prompt share, output length, bandwidth, and burst factor. Conditional routing earns its place as a separate candidate, not an unmeasured assumption.

The operational decision rule is equally clear. Stay aggregated when it already meets the SLO at lower accepted-token cost. Reserve fixed phase pools only when their advantage survives burst and mix sensitivity. Prefer conditional disaggregation when workload heterogeneity would otherwise strand one pool. Preserve rollback artifacts and rerun the crossover after material changes. That process turns “disaggregate or not” from an architectural preference into an auditable fleet-sizing decision.

Source links

Links cited in this article, in order of first appearance. Full addresses are provided for readers using extracted text.

1. https://arxiv.org/html/2401.09670v3
2. https://docs.nvidia.com/dynamo/reference/releases/v1-5-0
3. https://docs.nvidia.com/dynamo/dev/knowledge-base/concepts/system-architecture/disaggregated-serving
4. https://docs.vllm.ai/en/latest/features/disagg_prefill/
5. https://docs.nvidia.com/dynamo/dev/advanced-customizations/conditional-disaggregation
6. https://gpusmith.com/articles/en/llm-cost-per-million-tokens-benchmark
7. https://www.eia.gov/electricity/monthly/epm_table_grapher.php?t=epmt_5_6_a
8. https://docs.ray.io/en/latest/serve/llm/user-guides/prefill-decode.html
9. https://www.microsoft.com/en-us/research/publication/splitwise-efficient-generative-llm-inference-using-phase-splitting/
10. https://gpusmith.com/articles/en/llm-inference-hardware-sizing-guide
11. https://gpusmith.com/
12. https://docs.aws.amazon.com/sagemaker/latest/dg/sagemaker-hyperpod-model-deployment-dpd.html
13. https://docs.vllm.ai/en/latest/features/per_request_metrics/
14. https://docs.vllm.ai/en/latest/design/metrics/
15. https://docs.vllm.ai/en/stable/usage/metrics/
16. https://arxiv.org/html/2311.18677v2
17. https://arxiv.org/html/2607.02043v1
18. https://docs.nvidia.com/enterprise-reference-architectures/hgx-ai-factory-h100-h200-b200/latest/components.html
19. https://arxiv.org/html/2407.00079v4

20. https://nvidia.github.io/TensorRT-LLM/features/disagg-serving.html
21. https://nvidia.github.io/TensorRT-LLM/latest/developer-guide/kv-transfer.html
22. https://networking-docs.nvidia.com/connectx7hw/introduction
23. https://www.cisco.com/c/en/us/td/docs/unified_computing/ucs/c/hw/c885A/install/b_c885a-m8-server-hig/m_overview.html
24. https://docs.nvidia.com/deeplearning/triton-inference-server/user-guide/docs/user_guide/metrics.html
25. https://www.energy.gov/cmei/communicationstandards/style-guide-full-text
26. https://gpusmith.com/articles/en/gpu-cluster-energy-calculator
27. https://www.energystar.gov/ia/partners/prod_development/downloads/Data_Center_Metrics_Task_Force_Recommendations_V2.pdf
28. https://gpusmith.com/articles/en/gpu-cluster-reliability-budget
29. https://lenovopress.lenovo.com/lp2368-on-premise-vs-cloud-generative-ai-total-cost-of-ownership-2026-edition
30. https://arxiv.org/html/2401.11181v1
31. https://arxiv.org/html/2403.01876v1
32. https://www.nvidia.com/en-eu/data-center/h100/

THE PUBLISHER

About GPU Smith

GPU Smith provides independent engineering and research for private AI infrastructure. We help technical buyers and operators reason about GPU workload sizing, hardware selection, procurement, deployment and inference operations, from the compute system to the networking, power and cooling requirements around it.

Start with the workload

Useful infrastructure decisions begin with the models, data, throughput, latency and operating constraints a team actually has. GPU Smith helps connect those requirements with system design and deployment choices. The goal is a privately controlled AI environment whose capacity and operational demands are understood before a purchasing decision.

Hardware and supplier research

Our public reference library covers GPUs, complete systems and networking components. The server-vendor directory and manufacturing research help teams investigate suppliers, compare options and follow sources behind technical claims. We distinguish manufacturer specifications from measured benchmarks and advertised capabilities from tested configurations. Reference pages are research resources, not a live inventory listing or a binding equipment quote.

Deployment and operations

GPU Smith's engineering scope includes infrastructure integration, networking, power, cooling and the practical operation of inference workloads. Our published articles explain the tradeoffs behind deployment, procurement and ongoing operations so teams can ask better questions and document decisions.

Work with GPU Smith

Explore the hardware library, GPU references, networking references and vendor research. Contact GPU Smith with your workloads and deployment constraints to discuss a project and confirm current scope, pricing and availability.

Inclusion of a manufacturer or product in our research does not imply a partnership, certification or endorsement.

Website: https://gpusmith.com