GPU SMITH
COMPUTE ENGINEERING

GPU SMITH / RESEARCH & PRACTICAL GUIDANCE

GenAI-Perf to AIPerf Migration: Preserve Baselines

> 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 · 2026-10-13 · genai-perf to aiperf migration · aiperf · genai-perf · inference benchmarking · benchmark comparability · latency metrics · gpu acceptance · slo goodput · private ai

Original article: https://gpusmith.com/articles/en/aiperf-migration-preserve-genai-perf-baselines

In this article

1. Executive Summary
2. Introduction and Background
3. GenAI-Perf: The Baseline to Preserve
4. AIPerf: The Replacement to Qualify
5. Feature Comparison
6. Performance and Benchmarks
7. Migration and Acceptance Gate
8. Data Analysis and Evidence
9. Implications and Future Directions
10. Conclusion

Executive Summary

GenAI-Perf to AIPerf migration should be treated as a measurement change with a compatibility bridge. NVIDIA's **September 18, 2026** walkthrough identifies AIPerf as the ground-up successor to GenAI-Perf. (developer.nvidia.com) The migration guide also describes compatibility for currently supported features; these statements concern different things, architecture and supported interfaces. (docs.nvidia.com) As verified on **October 13, 2026**, the latest stable package is **AIPerf 0.13.0**, published September 24; subsequent development builds should not silently enter an acceptance test. (pypi.org) The practical sequence is to freeze the legacy evidence, qualify the new client, run both against an unchanged deployment, and authorize a new baseline only after reconciling the measurement definitions.

The largest semantic traps concern reasoning output and latency populations. When the endpoint returns reasoning separately in `reasoning_content` and the selected adapters handle that field as documented, compare legacy time to first token (**TTFT**) with AIPerf time to first output token (**TTFO**), and legacy output sequence length (**OSL**) with AIPerf Output Token Count. (docs.nvidia.com) (docs.nvidia.com) Reasoning embedded in ordinary `content`, such as `<think>` blocks, can remain in output counts unless filtered and may affect first-output timing. (docs.nvidia.com) Before applying this crosswalk, inspect a representative raw response or stream and verify each adapter's parsed fields, token counts and first-output event; model capability alone does not establish separation. Also examine inter-token latency (**ITL**) separately: the current AIPerf reference describes a token-normalized decode duration, whereas the legacy documentation describes intermediate-response gaps divided by tokens in the latter response. (docs.nvidia.com) (github.com) Streaming chunks can contain multiple buffered tokens, so a shared metric name cannot establish equivalence. (docs.vllm.ai)

Use separate regression, service-load and trace-replay profiles. Preserve rendered prompts, model and tokenizer revisions, actual output lengths, cache procedures, arrival timing, streaming settings and artifact provenance. Tokenizer revisions can identify commits, and container digests pin image content rather than a movable tag. (huggingface.co) (docs.docker.com) Increase client workers while holding server configuration

constant, then cross-check server counters and client resource pressure; a plateau alone does not prove server saturation. Prometheus rates require reset-aware handling, and Linux pressure measurements expose stalled tasks. (prometheus.io) (docs.kernel.org)

Acceptance should combine latency percentiles, error accounting, output throughput and **SLO-compliant goodput**, where SLO means service-level objective. AIPerf's good-request fraction keeps errors in its denominator. (docs.nvidia.com) This report provides a migration crosswalk, artifact manifest, traffic matrix and explicitly hypothetical calculations. Its confidence example uses the Wilson interval, not an invented hardware result. (www.itl.nist.gov) Procurement acceptance should require comparable workload evidence and correctness checks alongside speed, consistent with the separation of performance and accuracy in established inference benchmarking. (mlcommons.org)

Introduction and Background

Replacing a benchmark executable can alter the meaning of an apparently familiar chart. A procurement decision needs to distinguish changes in the server from changes in the client, dataset and accounting. This report examines **AIPerf vs GenAI-Perf** for private artificial intelligence (AI) platform engineers, performance teams and technical buyers who need to preserve historical large language model (LLM) inference evidence while adopting a new harness.

The relevant event is NVIDIA's successor walkthrough, rather than evidence that AIPerf first existed on that publication date. The release record separately identifies the current stable release and its focus on Kubernetes execution, speculative-decoding metrics and workload control. (github.com) Sources were fetched during this report's **October 13, 2026** research session. Mutable documentation should be treated as a dated reference, while the command example uses versioned documentation and an explicit package pin. (github.com)

The engineering perspective is acceptance against a written specification. GPU Smith describes engagements as using written acceptance criteria; it is an adjacent infrastructure advisor in this comparison, rather than another benchmark product. (gpusmith.com) The resulting recommendation is therefore a process for producing reviewable evidence, not a claim that changing tools makes a deployment faster.

Three evidence layers should remain distinguishable: the workload contract, the observed client events, and the serving system's counters. OpenAI's Chat Completions reference describes server-sent streaming events, while Prometheus documents counter-rate calculations. Neither substitutes for the other. (developers.openai.com) (prometheus.io) A timestamped client observation measures an experienced response; a server counter measures work recorded by that server. Local short-duration measurements can use a performance counter, whose documented purpose is measuring elapsed intervals at high resolution. (docs.python.org)

A fair comparison also requires stable execution identity. Kubernetes documents that image tags can move while digests are fixed; PyTorch cautions that identical seeds do not guarantee reproducibility across releases or platforms. (kubernetes.io) (docs.pytorch.org) Consequently, “same model, same seed” is an incomplete baseline description. The acceptance packet should explain what remained fixed, what changed deliberately, and which historical conclusions remain supportable.

GenAI-Perf: The Baseline to Preserve

Capabilities

GenAI-Perf provides the historical measurement contract that existing capacity and procurement decisions may reference. Its archived documentation defines output token throughput as benchmark output tokens divided by benchmark duration. (docs.nvidia.com) Retain that denominator explicitly. A normalized reporting layer should store numerator, interval and inclusion rules, rather than importing a column with a familiar name into a new dashboard.

The legacy implementation also documents ITL through intermediate responses and their token counts. (github.com) Preserve the raw responses and the implementation version wherever available. A mean, a per-request average and a pooled distribution of response gaps are different statistical objects, even when each is displayed in milliseconds.

Adoption

Before changing the harness, inventory the old baseline with the following controls:

- **Execution identity:** Capture the installed GenAI-Perf version, associated Perf Analyzer build, client operating environment and serving image digest. Docker documents digest-based image selection. (docs.docker.com)
- **Model identity:** Record weight revision, precision, runtime arguments and tokenizer commit or local directory. Hugging Face supports both revision selection and local vocabulary files. (huggingface.co) (huggingface.co)
- **Prompt identity:** Save rendered messages and chat templates, not only unrendered text. Models may use different control-token formats. (huggingface.co)
- **Sampling identity:** Archive the generated corpus and request ordering. GenAI-Perf documents reuse of synthetic payloads during benchmarking. (github.com)
- **Randomness:** Record the seed explicitly. The legacy documentation says an unspecified seed is randomized. (github.com)
- **Warmup and termination:** Recover warmup settings, measurement intervals, stabilization rules, request limits and stop reasons. Legacy warmup is separately configurable. (github.com)

When those artifacts are absent, classify the historical row as insufficiently specified. Recover the server digest from image-pull records when available; Docker prints it after pulling. (docs.docker.com) Pin the recovered image by digest rather than accepting the current value of its old tag. (docs.docker.com) Keep it for context, but establish a reproducible reference before allowing it to gate a hardware purchase.

Strengths and Limitations

The strength of a legacy baseline is continuity: it can tie present measurements to past decisions if its workload and accounting are recoverable. Its limitation is that continuity does not establish equivalence to a new metric definition. The official comparison matrix is a living document, and old stabilization controls cannot simply be assumed to retain their meaning in AIPerf. (docs.nvidia.com)

Repeated execution alone does not repair missing controls. PyTorch notes that algorithm selection can differ between runs because of benchmarking noise and hardware differences. (docs.pytorch.org) Record those runtime choices where relevant, and keep deterministic-operation settings fixed because they can also affect speed. (docs.pytorch.org) The objective is a defensible legacy reference, not an exact reproduction of an undocumented historical number. Controlled reference comparisons follow the same general principle as MLPerf's closed division, which holds the model and reference setup fixed. (docs.mlcommons.org)

AIPerf: The Replacement to Qualify

Capabilities

AIPerf uses a scalable multiprocess architecture and an extensible plugin system for endpoints, datasets, transports and metrics. (pypi.org) (pypi.org) Python documents subprocess-based multiprocessing as a way to bypass the Global Interpreter Lock. (docs.python.org) That architectural rationale supports testing client capacity; it does not independently prove that a particular client can sustain a target deployment's load.

The versioned dataset reference documents synthetic inputs, custom JSON Lines (JSONL) and public or trace-oriented formats, including ShareGPT, Mooncake and Baseten. (docs.nvidia.com) The repository additionally documents verbatim JSONL payload replay and multi-turn replay of saved inputs. (github.com) (github.com) Preserve the parser choice and the converted dataset checksum, because using the same dataset name is weaker than using identical serialized requests. Include the template's control tokens in this identity check because model formats can differ. (huggingface.co)

Adoption

Adopt the replacement in controlled steps:

- **Pin the package:** Use the stable version identified in the release record and retain the dependency lock or client container digest. (github.com) (docs.docker.com)
- **Prepare offline assets:** The repository documents local-tokenizer use without Hugging Face access. Preserve those assets with the model package for disconnected testing. (github.com)
- **Validate installation:** On Linux aarch64, the package documentation says the crick dependency requires a C compiler at installation time. Build that dependency into the approved client image. (pypi.org)
- **Validate the adapter:** Test representative serialized requests and parsed responses before generating acceptance load. Chat streaming uses incremental response chunks rather than a single complete response object. (developers.openai.com)
- **Archive repetition:** Keep individual runs and their summaries; the repository documents confidence intervals across repeated runs. (github.com)
- **Generation controls:** Verify supported stopping controls before enforcing a fixed-length workload. Backend support differs in the vLLM reference. (docs.vllm.ai)

- **Dependencies:** Freeze runtime dependency versions and algorithm-selection settings with the client image. Identical seeds do not guarantee cross-release reproducibility. (docs.pytorch.org)
- **Event accounting:** Record whether usage is obtained from the final stream chunk or reconstructed locally. (developers.openai.com)

The endpoint families include text generation, embeddings, ranking, audio and image-related workloads. Streaming metrics must be qualified per selected adapter: the CLI disables streaming for endpoint types that do not support it. (docs.nvidia.com) An embeddings result therefore cannot be assigned a meaningful token-stream ITL merely because another endpoint reports one.

Strengths and Limitations

The package's licensing is **Apache License 2.0**. (github.com) Operationally, the tool's breadth is useful only when the selected adapter preserves the old endpoint semantics. Inspect role handling, system prompts, output limits, reasoning fields, error responses and request controls.

Requested output lengths remain server-dependent. The published repository warns that generation-length constraints require supporting server controls. (github.com) In addition, final streaming usage statistics may be absent after interruption, according to the OpenAI reference. (developers.openai.com) Acceptance must therefore verify actual lengths and incomplete-stream handling, rather than treating configured means or missing usage fields as measured facts. Include accuracy checks in the performance run as well as in standalone quality testing. (mlcommons.org)

Feature Comparison

The migration crosswalk should be an executable review record: each row names the old interpretation, the proposed replacement and the evidence needed before comparison. *Table 1* summarizes the consequential differences. The official guide supplies the reasoning-token mapping; the versioned feature matrix supplies supported-interface boundaries. (docs.nvidia.com)

Baseline dimension	GenAI-Perf interpretation	AIPerf migration and acceptance evidence
Reasoning latency	TTFT starts at the first non-reasoning output token when reasoning is returned separately in `reasoning_content` and ignored by the legacy adapter.	Compare with **TTFO** only after a representative raw-response check verifies separate reasoning and documented adapter parsing. Reasoning embedded in ordinary `content` can affect first-output timing. Keep new TTFT as a separate series. (docs.nvidia.com) (docs.nvidia.com)
Reasoning token counts	Historical OSL excludes reasoning returned separately in `reasoning_content` when the legacy adapter ignores that field.	Compare with **Output Token Count** only when the raw-response check verifies separate-field parsing. Reasoning in ordinary `content`, including `<think>` blocks, remains counted unless filtered. Do not substitute total completion OSL. (docs.nvidia.com) (docs.nvidia.com)
ITL population	Intermediate-response gaps normalized by latter-response token counts.	Recompute a common statistic from raw events; current AIPerf ITL is a request-level normalized decode duration, normally (request latency - TTFT) / (OSL - 1). Lock token-count options as well as the formula. (docs.nvidia.com)

Baseline dimension	GenAI-Perf interpretation	AIPerf migration and acceptance evidence
Output throughput	Benchmark output tokens divided by benchmark duration.	AIPerf aggregate throughput uses total OSL divided by duration; reconcile reasoning inclusion. Its per-user rate is inverse ITL and excludes TTFT. (docs.nvidia.com)
Client parallelism	Global thread controls and legacy passthrough syntax.	Use native options and **workers-max**; retain **record-processors** settings and qualify result processing separately from request dispatch. (docs.nvidia.com)
Input payloads	Legacy generated input structure and prompt selection.	Convert payload to payloads and retain session/turn ordering. Validate converted requests against archived payloads. (docs.nvidia.com)
Telemetry	The old server-metrics-url points to GPU telemetry.	Map to **gpu-telemetry**, not the inference-server **server-metrics** surface. (docs.nvidia.com)
Stabilization and exports	Legacy windows, stability rules and comma-separated values (CSV)/JavaScript Object Notation (JSON) consumers.	Replace unsupported stabilization assumptions with explicit run rules; preserve the actual export schema and measurement interval. (docs.nvidia.com)
Length and stopping	Requested output limits and server-specific stopping.	Compare achieved lengths and stop reasons. Server support determines whether output constraints are enforced. (github.com)

The table deliberately marks semantic reconciliation as work, rather than claiming every flag substitution preserves a baseline. In particular, the migration guide's broad statement about unsupported analyze functionality is less specific than the current versioned matrix, which describes sweeps folded into profile. (docs.nvidia.com) Use the pinned matrix when porting sweep workflows, and retain a copy with the run packet.

A stronger endpoint check compares rendered messages before comparing tokens. Hugging Face documents conversion of structured messages into a formatted sequence, and warns against duplicating special tokens after template application. (huggingface.co) (huggingface.co) A different template can change the workload even if the visible human text looks identical.

Likewise, an application programming interface (API) usage field is not guaranteed to arrive with every stream event: OpenAI documents final-chunk usage reporting. (developers.openai.com) Decide whether the comparison uses server-reported counts or a pinned local tokenizer, record that choice, and reject a mixed-count series. Preserve raw errors and timing records, as the pinned vLLM benchmark reference also supports per-request exports. (docs.vllm.ai)

Performance and Benchmarks

Client saturation must be tested, not inferred from the architecture. Fix the deployment, corpus, arrival policy, cache procedure and client placement, then increase available request workers without changing the workload contract. Record worker placement and execution mechanism; Python distinguishes subprocess-based parallelism from threading. (docs.python.org) Keep concurrency high enough to permit the intended worker comparison; otherwise a worker cap can make nominal settings equivalent.

Use the following observations together:

- **Server work:** Collect an output-token counter or equivalent server-side work measure and record its exact definition. Prometheus rate reports per-second counter growth. (prometheus.io)
- **Counter handling:** Calculate rates before aggregation to preserve reset detection, and retain raw scrape timestamps. (prometheus.io)
- **Client scheduling:** Compare planned dispatch time with actual dispatch time. Python cautions that sleep can last longer than requested because of scheduling activity. (docs.python.org)
- **Client resources:** Capture host and client-cgroup pressure, not just a single central processing unit (CPU) utilization average. Linux exposes pressure information for cgroups as well as the host. (docs.kernel.org)
- **Transport pressure:** Inspect socket queues and memory with ss, and distinguish Transmission Control Protocol (TCP) queues from internal ZeroMQ pipes. (man7.org) (zguide.zeromq.org)

ZeroMQ's high-water marks describe internal pipe capacity; they are not a ready-made AIPerf health metric. Use exposed logs or instrumentation where available. Missing queue instrumentation should be recorded as an evidence gap, rather than converted into an assertion that queues were healthy. (zguide.zeromq.org)

Worker scaling is qualified when additional client capacity no longer changes server throughput beyond a predeclared tolerance, and the timing/resource evidence still shows that the target arrivals were delivered. A server plateau with growing client backlog remains inconclusive. Linux's pressure totals can reveal stalls that short averages obscure, while ss exposes socket statistics and queue columns. (docs.kernel.org) (man7.org) (man7.org)

For latency, record **p50, p95 and p99** as chosen acceptance reporting points, but define their populations first. AIPerf distinguishes ITL from the full distribution of consecutive inter-chunk arrival gaps. (docs.nvidia.com) vLLM's documented stream_interval can buffer multiple tokens before sending them. (docs.vllm.ai) Consequently, an event-gap p99 and a per-request decode-average p99 should remain separate series.

Server histogram percentiles also need a separate label: Prometheus interpolates quantiles within buckets. (prometheus.io) They are useful for correlation, but are not numerically interchangeable with percentiles calculated from retained client requests. Preserve bucket definitions when comparing telemetry across runs. (prometheus.io) Capture client and server timing independently before aligning their run windows. Elapsed-time counters do not provide an absolute cross-host timestamp. (docs.python.org) Use a monotonic elapsed-time source for local timing; Python documents that its monotonic clock is unaffected by system-clock updates. (docs.python.org)

No new hardware measurements were conducted for this report. The package's example output is itself labeled as not representing an official benchmark. (pypi.org) Neither that output nor NVIDIA's architectural explanation establishes an independently verified throughput improvement for a buyer's server, runtime or graphics processing unit (GPU).

Migration and Acceptance Gate

Start with a documented recipe and local smoke test, then perform the bridge on a deployment whose identity is fixed. Kubernetes uses the digest when an image specifies both tag and digest; record the resolved value, not just the human-readable tag. (kubernetes.io) Tokenizer loading can use a local vocabulary directory,

allowing the same assets to be preserved across both clients. (huggingface.co)

The following **regression recipe (Hypothetical Example)** uses documentation-checked AIPerf flags. It was not executed against an inference server. Replace MODEL and asset paths with the frozen deployment's values. The selected counts and lengths are illustrative test parameters, not vendor defaults. (docs.nvidia.com)

```
python -m pip install aiperf==0.13.0
aiperf profile \
  --model MODEL --url http://127.0.0.1:8000 \
  --tokenizer /approved/pinned-tokenizer \
  --endpoint-type chat --streaming \
  --synthetic-input-tokens-mean 512 \
  --synthetic-input-tokens-stddev 0 \
  --output-tokens-mean 128 --output-tokens-stddev 0 \
  --concurrency 8 --request-count 1000 \
  --warmup-request-count 32 --random-seed 42 \
  --workers-max 8 --artifact-dir artifacts/bridge
```

This generated fixture is useful for a smoke test or a new regression corpus. For the cross-tool bridge, substitute the frozen serialized dataset rather than relying on matching seed numbers. AIPerf documents verbatim payload replay, while GenAI-Perf documents dataset reuse. (github.com) (github.com) Verify the same rendered control tokens and archive request-level responses, errors and timing. (huggingface.co) (docs.vllm.ai)

The command requests an output-length distribution; it does not guarantee achieved OSL. Add server-supported generation controls only after checking payload acceptance. The vLLM benchmark reference illustrates that ignore_eos support differs across backends. (docs.vllm.ai) Also distinguish a harness seed from runtime determinism, which PyTorch does not guarantee across releases. (docs.pytorch.org)

Table 2 defines a recommended manifest. Store secrets separately and retain redacted commands.

Manifest group	Fields to archive	Acceptance purpose
Run identity	Run ID, Coordinated Universal Time (UTC) timestamps, baseline epoch, harness version and code revision	Identify the evidence packet without relying on directory names.
Serving identity	Image digest, runtime flags, model revision, precision, GPU topology and driver/runtime versions	Pin image content and document the deployment actually tested. (docs.docker.com)
Tokenization	Tokenizer commit or checksum, template, special-token policy and token-count source	Keep prompt formatting and counting reproducible. (huggingface.co)
Workload	Corpus checksum, rendered request sample, session/turn ordering, seed and achieved ISL/OSL distributions	Preserve serialized workload identity, not merely nominal lengths. (github.com)

Manifest group	Fields to archive	Acceptance purpose
Timing	Concurrency, arrival policy, schedule checksum, warmup, duration, stop rule and dispatch lateness	Make measured duration and arrival delivery reviewable; local monotonic timing avoids system-clock-update effects. (docs.python.org) (docs.python.org)
Client capacity	Worker cap, record-processing settings, CPU allocation, placement, pressure and socket observations	Show the generator could deliver the workload. (docs.kernel.org)
Outcomes	Complete, errored, timed-out and cancelled requests; raw events; latency population definitions	Permit independent denominator reconstruction. (docs.vllm.ai)
Exports	JSON/CSV artifacts, actual schema_version and aiperf_version, original config and integrity checksums	Validate consumers against the export actually produced. (docs.nvidia.com)

The export reference describes independent schema evolution for different output files. Do not infer every schema from the package version. Its telemetry naming description and version table also differ on which schema revision introduced the naming change; inspect the artifact and validate the parser instead of selecting a revision by assumption. (docs.nvidia.com)

Apply explicit pass/fail rules:

- **Serialization:** Representative rendered requests and parsed responses match the agreed endpoint contract. (huggingface.co)
- **Workload:** Achieved input and output distributions meet the frozen specification.
- **Load delivery:** Target arrivals are actually dispatched without unapproved concurrency throttling.
- **Client headroom:** Worker scaling and resource evidence pass the saturation qualification. (docs.kernel.org)
- **Latency:** The agreed p50/p95/p99 thresholds pass using reconciled event and sample definitions.
- **Goodput:** Joint request-level SLO compliance and error-inclusive attainment pass. Reject missing required metrics and check threshold units before evaluating the gate.
- **Correctness:** Output quality checks pass alongside speed, following the principle of accuracy verification in performance evaluation. (mlcommons.org)
- **Reproducibility:** A second controlled run meets the declared uncertainty and equivalence rules. (github.com)

Data Analysis and Evidence

A migration has two quantitative questions: whether the client delivered the intended load, and whether the delivered requests met the service contract. Throughput alone answers neither completely. Research such as DistServe defines goodput in relation to SLO attainment, while established inference evaluation makes latency constraints scenario-dependent. (arxiv.org) (mlcommons.org)

A concrete published specification illustrates why SLOs must be workload-specific. MLCommons' **March 24, 2026** GPT-OSS post describes the DeepSeek-R1 server scenario with Poisson arrivals, a **99th-percentile TTFT limit of 2 seconds** and a **time per output token (TPOT) limit of 80 milliseconds**. (mlcommons.org) These are constraints for that benchmark scenario. Enterprise acceptance thresholds should come from the service contract and retain the same metric definitions across the migration.

Table 3 separates recommended acceptance profiles. Here **ISL** means input sequence length and OSL means output sequence length. These are proposed engineering profiles, not measured benchmarks.

Profile	Workload and traffic contract	Pass/fail evidence
Fixed-length regression	Fixed rendered corpus and achieved ISL/OSL, matched concurrency and declared warm/cold cache procedure	Reconciled latency populations, output throughput, correctness and equivalence tolerance across both harnesses.
Service behavior	Preserved length distributions with Poisson arrivals or a documented gamma burstiness setting	Offered versus dispatched rates, errors, joint SLO attainment and client headroom. Record any binding concurrency ceiling.
Production-trace replay	Fixed timestamps, serialized payloads, sessions and cache reset/priming procedure	Schedule lateness, request coverage, latency by workload class and goodput for the replayed cohort.

These profiles measure different questions and should retain separate scores. AIPerf's timing reference says concurrency becomes a ceiling for rate-driven requests, while fixed-schedule mode uses dataset timestamps. (docs.nvidia.com) (docs.nvidia.com) Independent workload research also treats arrival processes and length distributions as explicit assumptions. (arxiv.org) Poisson and gamma controls should be stored with the original command, including the gamma smoothness setting. Keep the schedule immutable; vLLM's benchmark documentation similarly describes faithful replay based on timestamps. (docs.vllm.ai)

Client saturation calculation (Hypothetical Example)

Assume worker caps of **2, 4 and 8**, with server output rates of **8,000, 8,400 and 8,442 tokens/s**. Hold all server settings and workload parameters fixed. Define the relative worker-scaling change as:

```
delta = 100 * (new_server_rate - previous_server_rate) / previous_server_rate
```

The assumed changes are **5.00%** and **0.50%**. If a team predeclares **1%** as its plateau tolerance, the last change satisfies that arithmetic criterion. It still needs resource-pressure, socket and dispatch evidence before accepting client sufficiency. Compare wall time with process CPU time carefully: Python's process-time clock excludes sleep. (docs.python.org) Linux pressure exposes resource stalls, and ss provides socket statistics. (docs.kernel.org) (man7.org)

Compute the rates over aligned windows. Prometheus rate extrapolates across imperfect scrape alignment; increase can even return non-integer values for integer counters. (prometheus.io) (prometheus.io) Retain the raw counter series and explain the windowing method rather than rounding away a discrepancy.

SLO goodput calculation (Hypothetical Example)

Assume **1,000 attempted requests** during a **60-second** measured cohort: **920** satisfy every SLO, **40** complete but miss at least one, and **40** end in error. Successful completion throughput is **960 / 60 = 16.00 requests/s**; SLO-compliant goodput is **920 / 60 = 15.33 requests/s**. An attempted cohort must retain interrupted streams as outcomes even if final usage is absent. (developers.openai.com) The error-inclusive attainment fraction is **920 / 1,000 = 92.00%**, rather than the completed-only **95.83%**. AIPerf explicitly retains errors in this denominator. (docs.nvidia.com)

With an illustrative normal critical value of **1.96**, the Wilson **95%** interval for the assumed attainment fraction is approximately **90.15% to 93.53%**, using NIST's formula. (www.itl.nist.gov) This calculation assumes independent Bernoulli outcomes; a bursty trace may contain correlated requests. NIST also cautions about symmetric normal approximations when samples or failure counts are small. (www.itl.nist.gov) Use repeated independent runs or a correlation-aware analysis rather than treating the interval as proof of production reliability. The repository documents repeated-run confidence reporting, but that does not remove dependence within a trace. (github.com)

Apply request-level thresholds jointly, then calculate cohort attainment and rate. An all-threshold request gate differs from separately passing marginal percentiles. The vLLM reference likewise defines request-level goodput metrics for TTFT, TPOT and end-to-end latency. (docs.vllm.ai) DistServe's TPOT is a per-request token-generation average, reinforcing the need to state the sample population before comparing percentiles. (arxiv.org)

Implications and Future Directions

A successful migration separates the new measurement contract from the old operational conclusion. The acceptance owner should declare a **baseline epoch**, a versioned combination of workload, harness semantics, parser and deployment controls. New epochs make historical charts interpretable without implying that every break is a hardware regression or gain.

Use these decision rules:

- **Keep the legacy baseline:** Its inputs and accounting are recoverable, and the old accepted system remains unchanged. Preserve it as the historical record.
- **Rerun both harnesses:** A tool or parser changes while the server remains fixed. Require matched requests and reconciled metrics before linking the series.
- **Declare a new epoch:** Endpoint behavior, reasoning accounting, tokenizer/template, workload distribution, arrival policy, cache procedure or measurement population changes materially.

The identity controls are practical: Kubernetes documents how digest pinning avoids a registry change producing mixed image versions, and Hugging Face supports revision selection down to a commit. (kubernetes.io) (huggingface.co) Conversely, changing deterministic-operation settings can change measured speed, so include that configuration in the epoch. (docs.pytorch.org)

Rollback should restore the last qualified harness image, dataset and reporting parser together. Preserve the original artifact before conversion, and verify that the parser can still reconstruct the previously accepted denominator. Do not silently overwrite historical metrics with renamed fields. Pin restored images by digest so a later registry tag change cannot alter the rollback target. (kubernetes.io)

For private deployments, telemetry location matters as much as availability. The legacy documentation places Data Center GPU Manager (DCGM) Exporter on the inference-server machine. (github.com) AIPerf documents DCGM exporter collection and local pynvml GPU queries. (docs.nvidia.com) A local GPU-query backend on a separate client should not be assumed to observe remote server GPUs. Correlate deployment counters, GPU telemetry and client events within a declared measurement window; Python's elapsed-time counters require differences between readings, not cross-host subtraction. (docs.python.org)

Written evidence also supports future upgrades. GPU Smith's stated documentation scope includes architecture records, operating procedures and acceptance reports. (gpusmith.com) An effective migration packet applies that principle to benchmark commands, workload files and schema consumers. It should allow another engineer to reproduce the decision, rather than requiring trust in a screenshot. Established inference benchmarking uses a load generator for scheduling and latency tracking, underscoring that the client is part of the measurement system. (docs.mlcommons.org)

Conclusion

Preserving GenAI-Perf baselines requires a controlled bridge to AIPerf. Freeze the historical workload and serving deployment, qualify the replacement client, reconcile token and latency definitions, and compare both harnesses before approving a new reporting series. Treat architecture, interface compatibility and measurement equivalence as separate review questions.

The decisive artifact is a migration ledger tied to raw run evidence. It records what was preserved, which mappings were validated, where comparison remains conditional and what changed enough to require a new epoch. Reasoning-token handling, ITL populations, stream buffering, achieved output lengths and duration denominators deserve explicit review because each can change the interpretation of a reported improvement.

Acceptance should reflect the service being purchased or operated. A fixed-length regression profile protects continuity; an arrival-driven profile tests service behavior; a timestamped replay checks a captured workload. Each needs evidence that the client actually delivered its load, together with correctness, latency, errors, throughput and joint SLO attainment.

The worked examples in this report are arithmetic demonstrations, not deployment results. Teams should replace their assumptions with retained observations and uncertainty estimates before accepting a server, runtime or GPU. Keep the old baseline when its contract remains useful, rerun both tools when a bridge is possible, and declare a new epoch when the measurement contract materially changes. That sequence makes migration reviewable and preserves the meaning of historical capacity decisions.

Source links

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

1. https://developer.nvidia.com/blog/benchmarking-llm-inference-at-scale-with-aiperf/
2. https://docs.nvidia.com/aiperf/getting-started/migrating-from-gen-ai-perf
3. https://pypi.org/project/aiperf/
4. https://docs.nvidia.com/aiperf/v0.13.0/reference/ai-perf-metrics-reference
5. https://gpusmith.com/articles/en/llm-inference-metrics-ttft-itl-tps-goodput
6. https://github.com/triton-inference-server/perf_analyzer/blob/main/genai-perf/README.md
7. https://docs.vllm.ai/en/stable/api/vllm/config/scheduler/
8. https://huggingface.co/docs/transformers/main/en/model_doc/auto
9. https://docs.docker.com/reference/cli/docker/image/pull/
10. https://prometheus.io/docs/prometheus/latest/querying/functions/
11. https://docs.kernel.org/accounting/psi.html
12. https://docs.nvidia.com/aiperf/v0.13.0/tutorials/metrics-analysis/benchmark-goodput-with-ai-perf
13. https://www.itl.nist.gov/div898/handbook/prc/section2/prc241.htm
14. https://mlcommons.org/2026/03/mlperf-inference-gpt-oss/
15. https://github.com/ai-dynamo/aiperf/releases
16. https://gpusmith.com/
17. https://developers.openai.com/api/reference/resources/chat/subresources/completions/streaming-events
18. https://docs.python.org/3/library/time.html
19. https://kubernetes.io/docs/concepts/containers/images/
20. https://docs.pytorch.org/docs/main/notes/randomness.html
21. https://docs.nvidia.com/deeplearning/triton-inference-server/archives/triton-inference-server-2500/user-guide/docs/perf_analyzer/genai-perf/README.html
22. https://huggingface.co/docs/transformers/main/en/chat_templating
23. https://docs.nvidia.com/aiperf/v0.13.0/getting-started/gen-ai-perf-vs-ai-perf-cli-feature-comparison-matrix
24. https://docs.mlcommons.org/inference/submission/
25. https://docs.python.org/3/library/multiprocessing.html
26. https://docs.nvidia.com/aiperf/v0.13.0/reference/benchmark-datasets
27. https://github.com/ai-dynamo/aiperf
28. https://docs.vllm.ai/en/v0.31.0/cli/bench/serve/
29. https://docs.nvidia.com/aiperf/v0.13.0/reference/command-line-options
30. https://man7.org/linux/man-pages/man8/ss.8.html
31. https://zguide.zeromq.org/docs/chapter2/
32. https://docs.nvidia.com/aiperf/v0.13.0/reference/json-export-schema
33. https://arxiv.org/html/2401.09670v1
34. https://docs.nvidia.com/aiperf/v0.13.0/benchmark-modes/load-generator-options-reference
35. https://docs.nvidia.com/aiperf/v0.13.0/architecture-internals/architecture-of-ai-perf

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

This article is educational. Verify consequential medical, legal, financial, regulatory and technical decisions with appropriate professionals and the cited primary sources. Company services are described separately from the article's research findings.