



GPU Cluster Storage Sizing: Reads, Checkpoints, Recovery

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 · 2026-10-29 · gpu cluster storage sizing · gpu storage · storage capacity planning · checkpoint bandwidth · training data throughput · parallel file systems · checkpoint recovery · private gpu clusters

Original article: <https://gpusmith.com/articles/en/private-gpu-cluster-storage-sizing>



In this article

1. Executive Summary
 2. Introduction and Background
 3. Trace the Data Path and Define the Boundaries
 4. Build a Workload Inventory Before Choosing Capacity
 5. Calculate Read, Checkpoint and Network Envelopes
 6. Size Checkpoint Retention and the Recovery Path
 7. Turn the Workbook into Procurement Acceptance Tests
 8. Data Analysis and Evidence
 9. Implications and Future Directions
 10. Conclusion
-

Executive Summary

GPU cluster storage sizing starts with the bytes that jobs actually move, the time available to move them, and the failure boundary those bytes must survive. Specify separate envelopes for dataset reads, durable checkpoint writes, retained capacity and recovery. Cache location matters: NVIDIA describes local nonvolatile memory express (NVMe) storage as a place for temporary checkpoints before flushing to shared storage. Local completion therefore cannot, by itself, demonstrate shared-storage recovery readiness.

(docs.nvidia.com) Framework behavior also matters: **PyTorch** asynchronous checkpointing initially copies state into central processing unit (CPU) buffers, while **Orbax** can return before a save completes. (docs.pytorch.org) (orbax.readthedocs.io)

As of **October 29, 2026**, NVIDIA's B200 SuperPOD storage guidance, updated September 2, 2026, lists **40/20 GB/s** Standard and **125/62 GB/s** Enhanced aggregate read/write bandwidth per scalable unit (SU). (docs.nvidia.com) An SU contains **32 DGX B200 systems**. (docs.nvidia.com) These are reference-design guidelines, not minimums for every private cluster. The procurement calculation should begin with measured scan bytes, byte-weighted cache misses, overlapping jobs and actual committed checkpoints. Certification does not remove that work: NVIDIA grants storage certification to the exact product and configuration tested. (docs.nvidia.com)

The workbook below illustrates the calculation with an explicitly **hypothetical example**. Its inputs produce **8 GB/s** of warm-cache shared reads, **23 GB/s** during cold starts, and **25 GB/s** of synchronous checkpoint writes when **3 TB** must commit within **120 seconds**. Draining those checkpoints asynchronously over **600 seconds** instead requires **5 GB/s**, subject to staging capacity and backlog checks. These are calculated planning scenarios, not measured product results. The distinction is operational: Orbax waits for outstanding writes before starting a subsequent asynchronous save, while PyTorch recommends limiting asynchronous checkpoint requests to control memory pressure. (orbax.readthedocs.io) (docs.pytorch.org)

An effective request for proposal (RFP) asks vendors to reproduce the buyer's client count, file layout, access mode, durability boundary and concurrent workload. **MLPerf Storage** results are comparable within a workload, rather than across different workloads. (mlcommons.org) Its checkpoint test counts completion after file synchronization. (mlcommons.org) Use those methods as supporting evidence, then require application-

level pause and restart tests. Include metadata operations, sustained mixed traffic and a declared failure condition. **IOR**, **MDTest** and **fio** expose complementary filesystem, metadata and input/output (I/O) controls; none alone proves that a training job meets its recovery target. (ior.readthedocs.io) (www.vi4io.org) (fio.readthedocs.io)

Introduction and Background

The practical question is how much shared storage and network capacity an enterprise should buy before committing to a private graphics processing unit (GPU) cluster. A useful answer must explain which jobs can run together, how much useful training time checkpoints consume, and how quickly work can resume after a declared interruption. A capacity total without these conditions is an incomplete specification.

GPU Smith approaches this decision as an independent engineering advisor adjacent to the storage market. The report therefore uses public technical documentation and reproducible arithmetic rather than a preferred vendor configuration. It does not present the hypothetical workload as client experience, a benchmark or a delivery claim. The intended output is a workbook that storage leads, machine learning (ML) platform architects and procurement teams can inspect together.

The data path deserves attention before the storage bill of materials. NVIDIA's DGX GB documentation describes shared storage as the original data source and local NVMe as an optional cache. (docs.nvidia.com) **TensorFlow** separately documents application caching and prefetch buffers. (www.tensorflow.org) These mechanisms operate at different places in the path. A cache-hit figure is useful only when its boundary is named and the remaining traffic can be traced to a specific storage service.

Similarly, a model export and a resumable training checkpoint are different planning objects. **DeepSpeed** saves model, optimizer and learning-rate scheduler states; **Lightning** describes broader internal training state. (www.deepspeed.ai) (lightning.ai) Procurement should use the complete bytes written by the intended training recipe, rather than estimate solely from model parameter count.

All external documentation cited here was retrieved on **October 29, 2026**. Mutable documentation should be paired with pinned software versions in the eventual acceptance record. The equations and acceptance thresholds are engineering proposals. Their numerical example is explicitly hypothetical, so its assumptions can be replaced without confusing them with published measurements.

Trace the Data Path and Define the Boundaries

Source, cache, worker and checkpoint sink

Draw the path from the authoritative dataset to the storage client, application worker and accelerator. Then draw the return path from training state through any staging tier to the checkpoint repository. Record which component owns each copy and which failures invalidate it. NVIDIA's reference architecture describes a high-performance, Portable Operating System Interface (POSIX)-style filesystem supporting operations across nodes; that is an architectural example rather than proof that every candidate exposes identical semantics. (docs.nvidia.com)

Streaming changes the path, too. **Hugging Face Datasets** can iterate over a dataset without first downloading the complete collection. (huggingface.co) Its streaming shuffle uses a configurable buffer. (huggingface.co) A buyer using streaming must trace traffic to its actual source. It would be misleading to allocate those reads entirely to a local parallel filesystem if the worker instead reads from a remote object endpoint.

Use the following boundary record in the workbook:

- **Authoritative source:** name the dataset version, owning service and required availability.
- **Cache tier:** identify application memory, operating-system cache, local NVMe or storage-server cache.
- **Read endpoint:** identify the client mount, object API or other interface the training process uses.
- **Checkpoint staging:** identify where an immutable snapshot waits before its durable commit.
- **Commit boundary:** state what completion means and which failures the committed copy must survive.
- **Restore boundary:** identify which surviving service can provide a complete restartable generation.

File synchronization is relevant but should be interpreted precisely. Linux documents that `fsync` waits until the device reports completion. (www.man7.org) It also distinguishes synchronizing file contents from making the containing directory entry durable. (www.man7.org) Require the checkpoint library and storage supplier to explain the complete publication sequence, including its manifest or equivalent completion record. Do not infer power-loss resilience solely from the duration of an application write call.

Direct paths are configuration requirements

GPUDirect Storage (GDS) enables direct memory access between storage and GPU memory without a CPU bounce buffer, but also has a compatibility path through CPU memory. (docs.nvidia.com) The RFP should identify the path actually exercised and require evidence from the proposed configuration. A label on a datasheet is insufficient to establish how a particular application transfers bytes.

IBM Storage Scale documents alignment and access-coordination requirements for direct I/O. (www.ibm.com) **fiio** describes its direct option as requesting non-buffered I/O, usually through `O_DIRECT`. (fio.readthedocs.io) These are related controls, not interchangeable promises. State the application API, filesystem, client version, buffer alignment and instrumentation used to confirm the path. If the production job uses buffered reads, preserve that behavior in the application acceptance test and run a separately identified cache-avoidance test for backend characterization.

Build a Workload Inventory Before Choosing Capacity

Collect a trace with provenance

The workbook should contain a row for each job class and a separate schedule showing permitted overlap. Include training, fine-tuning, evaluation, dataset preparation and checkpoint restore. Do not assume that their busiest periods are mutually exclusive unless a scheduler policy enforces that condition.

Table 1 defines the minimum input sheet. Each value should carry an owner, collection date, observation window and measurement boundary. Entries marked as assumptions remain assumptions until a trace or acceptance run replaces them.

Input	Meaning and units	Preferred provenance
Dataset scan bytes	Bytes consumed during the declared read window, before the cache being evaluated	Application trace tied to dataset version
Byte-weighted hit rate	Fraction of requested bytes served before shared storage	Matched application and client counters
Read window	Seconds over which the bytes must arrive	Training schedule and batch timing
Concurrent jobs	Jobs allowed to overlap, including cold starts and restarts	Scheduler policy plus observed timeline
Checkpoint generation	Actual total bytes in a complete resumable save across ranks	Completed generation inventory
Pause and drain budgets	Permitted foreground stall and background completion windows, in seconds	Service objective and framework trace
Interval and retention	Checkpoint schedule and number of complete retained generations	Recovery policy and measured save behavior
Dataset and working capacity	Unique retained datasets, transformations, staging and operational reserve	Namespace inventory and lifecycle rules
Recovery target	Elapsed seconds from the declared incident boundary to useful work	Restart exercise with component timestamps

The table deliberately separates logical consumption from shared-service demand. A high request-hit rate can coexist with substantial missed bytes if the missed requests are larger. Use byte-weighted hits for the throughput equation and keep request counts for metadata and latency analysis. This is a measurement definition for the workbook, not a claim about any particular cache implementation.

TensorFlow documents that caching avoids repeated upstream work. (www.tensorflow.org) Its prefetching uses a background thread and buffer. (www.tensorflow.org) Accordingly, collect both cold and warm executions, record the cache placement, and track memory occupied by buffering. Hugging Face streaming also changes shard order during shuffling, reinforcing the need to preserve the production access pattern in the trace. (huggingface.co)

Collect the following alongside aggregate bytes:

- **File-size distribution:** separate large checkpoint shards from small training samples.
- **Operation mix:** count opens, creates, stats, renames and deletes as well as data transfers.
- **Concurrency:** record active clients, reader processes, workers and outstanding requests.
- **Tail behavior:** preserve slow batches and slow checkpoint completions instead of only averages.
- **Traffic boundary:** distinguish logical bytes, filesystem bytes and bytes observed on network ports.
- **Retention transitions:** capture temporary overlap while a new generation is committed and an old one is deleted.

For metadata testing, **MDTest** explicitly measures filesystem metadata performance through a synthetic parallel workload. (www.vi4io.org) Use it to isolate that dimension, then confirm whether the real job has the same directory layout and operation mix. A headline bandwidth figure cannot substitute for this check.

Calculate Read, Checkpoint and Network Envelopes

Dataset read throughput requirements

For job j , let B_j be requested bytes during window T_j and h_j the byte-weighted hit rate before shared storage. Then $R_j = B_j \times (1 - h_j) / T_j$. Sum overlapping jobs and measured ingestion, evaluation and restore traffic using the same resources. This is an accounting identity, not a vendor benchmark.

Keep warm-cache, cold-start and synchronized-start cases. If the trace measures shared-storage misses directly, do not apply the cache adjustment again. Distinguish compressed storage bytes from expanded tensors at the decompression boundary.

Amazon FSx for Lustre illustrates bottleneck separation: cached reads can be network-limited. (docs.aws.amazon.com) Uncached reads and writes depend on the lower of network and disk throughput. (docs.aws.amazon.com) Its Persistent 2 deployment permits independently provisioned metadata I/O operations per second (IOPS). (docs.aws.amazon.com) Measure these layers privately; do not import the service settings.

Checkpoint write bandwidth calculation

For a synchronous save, define C as the complete job-level bytes that must commit and P as the permitted training pause. $W_{\text{sync}} = C / P$ is a lower-bound planning rate when that whole pause is available for storage. If state extraction, serialization or coordination consume part of P , subtract their non-overlapped time before dividing. NVIDIA's storage guidance explicitly notes that synchronous checkpoint activity stops training. (docs.nvidia.com)

For an asynchronous save, use $W_{\text{drain}} = C / D$, where D is the allowed background drain window. Separately measure the foreground snapshot or staging pause. PyTorch's recipe notes CPU-buffer copies and additional memory requirements. (docs.pytorch.org) It recommends limiting outstanding asynchronous requests. (docs.pytorch.org) Calling a save asynchronous therefore does not establish that all checkpoint overhead disappears.

For a shared drain queue, a useful bookkeeping recurrence is $Q_{\text{next}} = \max(0, Q_{\text{now}} + C_{\text{arriving}} - W_{\text{available}} \times \Delta t)$. Apply it over the concurrent-job timeline, with available bandwidth reduced by competing work. Long-run write demand must remain below the available sustained service rate, and staging must hold the largest backlog. Orbx's documented wait for outstanding saves before starting another illustrates how a missed drain deadline can become a later foreground delay. (orbx.readthedocs.io)

Network capacity and mixed traffic

Payload gigabits per second equal decimal gigabytes per second multiplied by 8. (physics.nist.gov) Divide required payload by measured end-to-end efficiency before comparing nominal link rates. Identify whether each calculation concerns a client port, storage uplink or aggregate network cut.

Do not add opposite directions blindly across a full-duplex port, and do not assume read and write maxima are simultaneously available from a shared controller. Request mixed-workload measurements. **iperf3** offers parallel streams and simultaneous bidirectional testing, which can characterize network behavior separately

from storage. (software.es.net) Storage acceptance must still exercise the filesystem path.

- **Sustained envelope:** calculate full-window read and write demand for each allowed overlap.
- **Burst envelope:** include synchronized checkpoint arrivals, cold-cache starts and restores.
- **Client envelope:** check each node's ports and client throughput, not only aggregate totals.
- **Shared envelope:** test controller, target and uplink resources with concurrent reads and writes.
- **Efficiency evidence:** measure protocol and topology effects on the offered configuration.
- **Percentile discipline:** compare demand distributions with matching observation windows, not unrelated percentiles.

Size Checkpoint Retention and the Recovery Path

Capacity follows complete generations

Begin with unique retained datasets, prepared copies, complete checkpoint generations, active staging and operational reserves. For uniform checkpoint generations, **retained checkpoint capacity = generation bytes × retained generations**. For mixed jobs, sum each job's own size and policy. Include the generation being written if it must coexist with all retained generations; identify whether the retention count already includes it.

Use completed training checkpoints for that inventory. DeepSpeed's saved states include more than model weights. (www.deepspeed.ai) **Hugging Face Accelerate** identifies optimizer, random number generator (RNG) and gradient scaler state alongside the model. (www.deepspeed.ai) (huggingface.co) Lightning similarly distinguishes loading weights from restoring full training. (lightning.ai) DeepSpeed also provides client state for additional application data. (www.deepspeed.ai) Measure the actual recipe because these descriptions do not establish a universal bytes-per-parameter multiplier.

If L is the planned live footprint and f is the required free fraction of provisioned usable space, the workbook equation is **usable provisioned capacity = $L / (1 - f)$** . The free fraction is a buyer assumption to test, not a universal filesystem requirement. Keep replication, erasure coding, parity, spares and formatting overhead in a separate vendor-specific usable-to-raw conversion. Applying an assumed protection multiplier to a quote already expressed in usable capacity would double count it.

Recovery time is an end-to-end ledger

Define the recovery time objective (RTO) from a specific event to the first useful resumed training step. Budget detection, scheduling, node allocation, checkpoint selection, reads, deserialization, redistribution, data-loader preparation and validation. Overlapping stages should be represented as an actual timeline; a simple sum is conservative only when its assumed sequencing matches the implementation.

PyTorch Distributed Checkpoint supports parallel operation across ranks. (docs.pytorch.org) It also supports load-time resharding between cluster topologies. (docs.pytorch.org) DeepSpeed documents conversion to Universal format for its corresponding checkpoint workflow. (www.deepspeed.ai) These features motivate explicit restart scenarios, including changed worker counts, but do not establish that every version or saved format will restore interchangeably.

Record these recovery requirements:

- **Compatible state:** pin framework, format, training script and relevant dependencies.
- **Complete generation:** select the latest demonstrably committed save rather than a partial directory.
- **Changed topology:** exercise the permitted replacement-node and rank configuration.
- **Usable progress:** validate resumed state and a completed training step, not merely successful file reads.
- **Independent repository:** restore from a source that survives the failure being tested.
- **Measured loss:** record the age of the last usable checkpoint and resulting lost work.

TensorFlow gathers checkpoint variables from reachable tracked objects. (www.tensorflow.org) Its `assert_consumed` checks whether the program and checkpoint match exactly. (www.tensorflow.org) Accelerate states that its save and load states should come from the same training script, and PyTorch documents limits on backward compatibility for distributed checkpoint state dictionaries. ([huggingface.co](https://huggingface.co/docs/pytorch.org)) The RFP should therefore treat software compatibility and storage restore speed as separate acceptance conditions.

Turn the Workbook into Procurement Acceptance Tests

Match the filesystem configuration to the workload

Lustre distributes files across object storage targets and recommends increasing striping for a specific performance need. (wiki.lustre.org) **BeeGFS** supports per-directory and per-file striping. (doc.beegfs.io) Directory changes affect new files. (doc.beegfs.io) Record effective layouts and recreate test files where required.

Require documented POSIX, namespace, durability and direct-I/O semantics. IBM documents application coordination requirements. (www.ibm.com) NVIDIA-Certified Storage scopes approval to the tested configuration and evaluates sustained behavior and latency. (docs.nvidia.com)

Table 2 translates the workbook into procurement requirements. The acceptance thresholds are buyer-defined variables until the workload trace supplies values.

Requirement	Evidence requested	Acceptance condition
Dataset delivery	Exact client count, file layout, cache state and concurrent jobs	Meets calculated cold and warm read envelopes with acceptable batch stalls
Synchronous checkpoint	Committed bytes, timing breakdown and completion boundary	Full save completes inside the permitted pause
Asynchronous checkpoint	Staging footprint, drain trace, queue depth and completion signal	Foreground pause and drain deadline met without growing backlog
Shared capacity	Usable and raw accounting, retention transitions and fill level	Retains required complete generations and declared free reserve
Metadata	Production-like directory and file-operation mix	Required operation rate and latency under concurrent data traffic

Requirement	Evidence requested	Acceptance condition
Network path	Port mapping, oversubscription and end-to-end measurements	Each relevant path carries the declared mixed-workload envelope
Recovery	Pinned software, declared failure condition and full restart timeline	Resumes useful training within RTO with validated state
Reproducibility	Test scripts, configuration, counters and repeated-run results	Buyer can reproduce results on the supplied configuration

Request source, configuration date and units for each result. Keep vendor-quoted sustained read/write GB/s, observed 95th-percentile demand, lowest sustained service windows, pause percentiles and restore duration separate. Record observation windows; high demand does not establish available throughput.

Use complementary tools, then test the application

IOR supports synchronized POSIX write close. ([ior.readthedocs.io](https://www.readthedocs.io/en/latest/ior/)) It also alters readback ordering. ([ior.readthedocs.io](https://www.readthedocs.io/en/latest/ior/)) fio exposes direct-I/O and synchronization controls. ([fio.readthedocs.io](https://www.readthedocs.io/en/latest/fio/)) **DAOS** distinguishes IOR POSIX from its native filesystem interface. ([docs.daos.io](https://www.readthedocs.io/en/latest/daos/)) It documents client processes controlled through Message Passing Interface (MPI). ([docs.daos.io](https://www.readthedocs.io/en/latest/daos/)) Match benchmark and application interfaces.

For tail measurements, fio offers histogram logging to retain percentile accuracy. ([fio.readthedocs.io](https://www.readthedocs.io/en/latest/fio/)) Preserve those distributions rather than deriving tail latency from interval averages.

Run the following acceptance sequence:

- **Reproduce the bill of materials:** record clients, storage nodes, media, switches and software versions.
- **Establish cold and warm states:** document precisely which caches remain populated.
- **Measure bulk transfers:** use production-like shard sizes, access patterns and client counts.
- **Measure metadata separately:** exercise expected namespace operations and directory scale.
- **Mix the jobs:** overlap reads, checkpoint writes, ingestion and agreed competing workloads.
- **Observe complete windows:** preserve per-job timing, tails, queue depth and committed bytes.
- **Exercise the failure condition:** test the agreed surviving path and restore procedure.
- **Repeat without selective reporting:** retain configuration changes and all completed-run results.

NetApp ONTAP documents competing-workload throughput ceilings. ([docs.netapp.com](https://www.readthedocs.io/en/latest/docs.netapp.com/)) Insufficient headroom can constrain floors. ([docs.netapp.com](https://www.readthedocs.io/en/latest/docs.netapp.com/)) Test proposed quality of service (QoS) policies under contention.

Data Analysis and Evidence

Published reference points and their limits

NVIDIA's September 2026 B200 table gives **40/20 GB/s** Standard and **125/62 GB/s** Enhanced reads/writes per SU. ([docs.nvidia.com](https://www.readthedocs.io/en/latest/docs.nvidia.com/)) An SU contains **32 systems**, with separate storage and compute fabrics. ([docs.nvidia.com](https://www.readthedocs.io/en/latest/docs.nvidia.com/)) Compare only after matching workload and topology.

MLPerf Storage requires training datasets at least **five times aggregate client memory**. (mlcommons.org) Runs require **500 batches per accelerator**. (mlcommons.org) Checkpoint runs comprise **10 writes followed by 10 reads**. (mlcommons.org) Repository guidance excludes transfer into accelerator memory, but carries a different version label from current results. (github.com) Preserve benchmark versions and avoid combining release rules.

Auditable workbook: hypothetical example

All inputs and outputs in this example are **hypothetical planning assumptions**, not supplier specifications or observations. Decimal units apply: **1 TB = 1,000 GB**, and **1 GB = 1,000,000,000 bytes**. (physics.nist.gov) Jobs A and B each scan **1.2 TB in 120 seconds**, with a **75% byte-weighted hit rate** before shared storage. Job C continuously reads **3 GB/s** from shared storage. A and B checkpoint **2 TB** and **1 TB**, respectively, on a **900-second start-to-start schedule**. Assume their checkpoints overlap.

Table 3 makes each hypothetical output reproducible from the stated assumptions.

Workbook output	Calculation using hypothetical inputs	Derived requirement
Warm shared reads	$2 \times (1,200 \times 0.25 / 120) + 3$	8 GB/s
Cold shared reads	$2 \times (1,200 / 120) + 3$	23 GB/s
Synchronous checkpoint writes	$(2,000 + 1,000) / 120$	25 GB/s if the entire pause is storage time
Warm reads plus synchronous writes	$8 + 25$	33 GB/s aggregate service demand, with directions identified
Cold reads plus synchronous writes	$23 + 25$	48 GB/s aggregate service demand if that overlap is allowed
Asynchronous drain	$3,000 / 600$	5 GB/s while draining; 13 GB/s with warm reads
Average checkpoint ingress	$3,000 / 900$	3.33 GB/s , rounded; lower than the drain requirement
Retained checkpoint payload	$3 \text{ TB} \times 3 \text{ retained generations}$	9 TB
Provisioned usable capacity	$(240 + 9 + 3 + 18) / 0.80$	337.5 TB , before vendor-specific raw conversion

Capacity assumes **240 TB** datasets, **9 TB** retained checkpoints, **3 TB** additional in-flight staging and **18 TB** other live space: **270 TB** total. The assumed **20% free fraction** yields the provisioned result. Retention excludes the in-flight generation. If staging resides on clients, move its allowance there. Budget actual client memory or NVMe staging separately.

Warm synchronous payload is **64 Gb/s** read-direction and **200 Gb/s** write-direction. Assuming **70% efficiency**, nominal budgets are **91.4 Gb/s** and **285.7 Gb/s**. An assumed **25% demand margin** raises them to **114.3 Gb/s** and **357.1 Gb/s**. Keep full-duplex directions separate; calculate cold starts independently. These are assumptions, not topology measurements. iperf3 supports bidirectional testing. (software.es.net)

For both jobs, assume a **900-second RTO** and **10 GB/s** aggregate restore reads. Budget detection/scheduling **120 seconds**, allocation **60**, reading **3 TB 300 seconds**, deserialization **120**, redistribution **60**, and first-batch preparation **30**. Total: **690 seconds**, leaving **210 seconds**. Replace every allowance with a trace through useful training progress.

Implications and Future Directions

The central procurement implication is that different workload changes affect different dimensions of storage. A better warm-cache hit rate reduces repeated shared reads, but it does not reduce the bytes in a checkpoint. A shorter permitted checkpoint pause increases the required foreground write rate unless the implementation moves work into staging or overlaps it with training. These are consequences of the workbook equations, not forecasts of any vendor's performance.

For asynchronous designs, demand evidence about the transition from foreground pause to durable completion. PyTorch's CPU-buffer staging and Orbx's explicit completion wait illustrate distinct events in that transition. (docs.pytorch.org) (orbax.readthedocs.io) A platform roadmap that mentions asynchronous saving should therefore trigger new measurements of memory, staging occupancy and subsequent-save behavior, rather than an automatic reduction in the storage requirement.

The same discipline applies to application and infrastructure upgrades. Hugging Face documents a buffered streaming shuffle, while TensorFlow documents caching and prefetching. (huggingface.co) (www.tensorflow.org) Changing a pipeline can change where bytes move even when the model is unchanged. Preserve a versioned workload trace with the training recipe and rerun the relevant acceptance case when its I/O behavior changes.

Future procurement comparisons should retain **configuration-specific evidence**. NVIDIA's certification scope is exact-configuration based, and MLPerf Storage warns against comparing different workloads as equivalent results. (docs.nvidia.com) (mlcommons.org) Request reproducible results for the actual client and network arrangement, rather than extrapolating an unrelated topology.

For private clusters, this also favors acceptance packages that can run inside the intended operating environment. Keep scripts, prepared test data, configuration manifests and analysis locally available. DAOS's documentation of separate IOR backends and MDTest's metadata focus illustrate why the package should state what each test covers. (docs.daos.io) (www.vi4io.org) The storage engineering team can then revisit one changed assumption without rebuilding the entire procurement argument.

Conclusion

Size shared GPU-cluster storage from a workload ledger with explicit units and failure boundaries. Start with bytes requested during a real read window, remove only the byte-weighted hits served before the storage service, and add traffic from jobs that can overlap. Preserve cold-start and recovery cases alongside the warm steady-state calculation. The resulting envelope should describe both the aggregate service and the path available to individual clients.

Checkpoint sizing requires two clocks: the time training is paused and the time until the complete generation is durably available. Divide committed bytes by the appropriate window, then account for staging, background contention and backlog. Capacity planning must retain datasets, complete generations and in-flight work without double counting protection or free-space allowances. Recovery planning must extend through validated training progress rather than stop at the last read.

The hypothetical workbook makes these choices reviewable, but its inputs are replaceable assumptions. Its value is the chain from input provenance to equation, resulting requirement and acceptance evidence. A buyer can change the permitted overlap, checkpoint interval or recovery target and see which part of the specification must change.

An actionable RFP therefore includes the workload inventory, read/write envelopes, usable-capacity accounting, restart timeline and reproducible test matrix. Vendor quotations become comparable when they use those same boundaries. The final purchase decision can then rest on measured application outcomes and a configuration the buyer can reproduce.

Source links

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

1. <https://docs.nvidia.com/dgx/dgxgb200-user-guide/storage.html>
2. https://docs.pytorch.org/tutorials/recipes/distributed_async_checkpoint_recipe.html
3. https://orbax.readthedocs.io/en/latest/guides/checkpoint/async_checkpointing.html
4. <https://docs.nvidia.com/dgx-superpod/reference-architecture-scalable-infrastructure-b200/latest/storage-architecture.html>
5. <https://docs.nvidia.com/dgx-superpod/reference-architecture-scalable-infrastructure-b200/latest/dgx-superpod-components.html>
6. <https://docs.nvidia.com/certification-programs/certified-storage/latest/nvidia-certified-storage.html>
7. <https://mlcommons.org/benchmarks/storage>
8. <https://ior.readthedocs.io/en/latest/userDoc/options.html>
9. <https://www.vi4io.org/tools/benchmarks/mdtest>
10. https://fio.readthedocs.io/en/master/fio_doc.html
11. https://www.tensorflow.org/guide/data_performance
12. <https://www.deepspeed.ai/getting-started>
13. https://lightning.ai/docs/pytorch/stable/common/checkpointing_basic.html
14. <https://huggingface.co/docs/datasets/en/stream>
15. <https://www.man7.org/linux/man-pages/man2/fsync.2.html>
16. <https://docs.nvidia.com/gpudirect-storage/overview-guide/index.html>
17. <https://www.ibm.com/docs/en/storage-scale/6.0.1?topic=applications-considerations-use-direct-io-o-direct>
18. <https://docs.aws.amazon.com/fsx/latest/LustreGuide/performance.html>
19. <https://physics.nist.gov/cuu/Units/binary.html>
20. <https://software.es.net/iperf/invoking.html>

21. https://huggingface.co/docs/accelerate/usage_guides/checkpoint
22. <https://docs.pytorch.org/docs/main/distributed/checkpoint.html>
23. <https://www.deepspeed.ai/tutorials/universal-checkpointing>
24. <https://www.tensorflow.org/guide/checkpoint>
25. https://wiki.lustre.org/Lustre_Striping_Best_Practices
26. https://doc.beegfs.io/7.4.6/advanced_topics/striping.html
27. <https://docs.daos.io/latest/testing/ior>
28. <https://docs.netapp.com/us-en/ontap/concepts/load-balancing-concept.html>
29. https://github.com/mlcommons/storage/blob/main/Submission_guidelines.md

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.