CLOCK WORK

The state of AI snapshots today

November 13, 2026 · clockwork.io/blog/the-state-of-ai-snapshots-today/

Nov 12, 2026

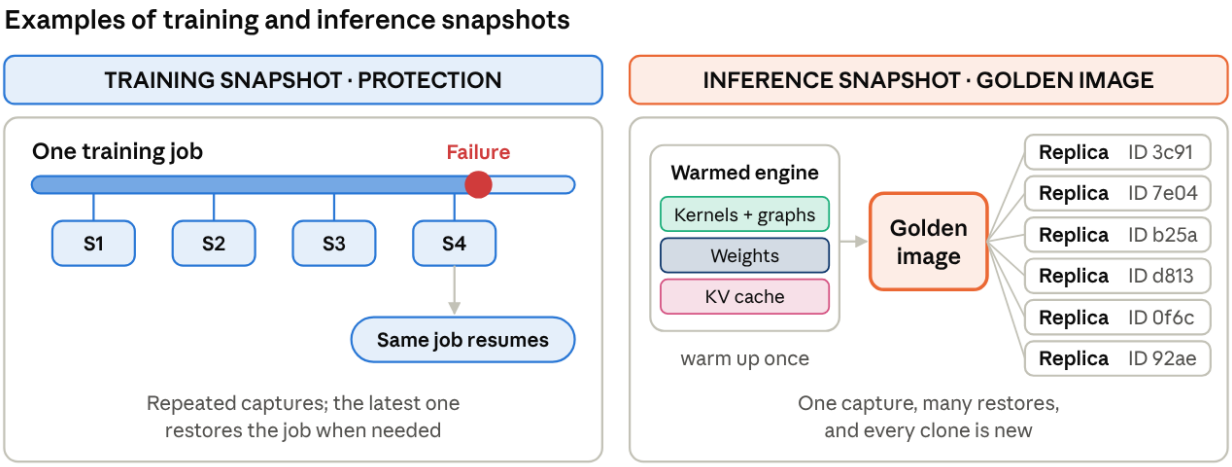
This post is inspired by NVIDIA's announcement this week, which puts snapshots in the spotlight and highlights their value in speeding-up inference cold starts. The results Nvidia published are impressive: an 8-36x improvement in startup time vs a cold start. Snapshots also have a broader role in AI, including protecting long-running training jobs. This post covers the current state of snapshots for both inference and training: what works and what has yet to be solved.

A snapshot saves the execution state of a running program so it can resume later from the same point, often on a different machine. For GPU workloads, that means the running host process and everything on the GPU, including model weights in HBM and CUDA state. The host process is typically captured with Linux CRIU (checkpoint and restore in userspace). GPU state can be captured with NVIDIA's cuda-checkpoint utility, which copies device memory to host memory and reconstructs it on restore. Note that snapshots should not be confused with application checkpoints, which record logical state such as weights and optimizer state, and require fresh runtimes to load them (our earlier post covers this distinction in more depth).

Training and inference are two examples of where snapshots are used in AI, each for a different purpose. A training snapshot protects a running job at an exact point in time so it can later be resumed (e.g. after failure or preemption). An inference snapshot is a golden image: it is taken once from a warmed-up engine before it begins serving, and restored many times to start new replicas

© 2026 CLOCKWORK. All rights reserved.

without repeating compilation and graph capture.



training snapshot vs inference snapshot

Inference: what is available today

Inference snapshots now work in production for a single GPU and, as of this week in pre-release form, for up to eight GPUs on a single node.

Modal described its approach in May 2026. It combines gVisor checkpoint/restore for host state with the NVIDIA driver's device checkpointing for GPU state. Modal reports typical speedups of 4–10x. Impressively, Modal restored about 15 million replicas from CPU and GPU snapshots between February and April 2026. That is strong evidence of snapshot use at production scale. GPU snapshots remain in alpha, and Modal's documentation calls them "generally incompatible with multi-GPU code."

© 2026 CLOCKWORK. All rights reserved.

NVIDIA Snapshot, announced November 10, is an open-source checkpoint and restore system for GPU workloads on Kubernetes. It was originally developed for NVIDIA Dynamo inference workers and is now a component other engines and platforms can use. It ships examples for vLLM, SGLang and TensorRT-LLM. Its main advance is scope: a workload spanning multiple GPUs on one node is captured as a single artifact. On B200 GPUs, NVIDIA measured restores of 3.6 to 42.7 seconds, 8x to 36x faster than a cold start. The largest eight-GPU models restored in under 43 seconds, against cold starts of 6 to 19 minutes. These results are very impressive!

Partners report similar results. Crusoe, an upstream contributor, cut GLM-5.3 pod readiness from about 8 minutes to 31 seconds on HGX B300. Saturn Cloud measured GLM 5.3 on eight B300 GPUs going from about 8.5 minutes to 34 seconds.

The wider field is active too. Google's GKE Pod snapshots save CPU and GPU memory for Pods running in GKE Sandbox, with multi-GPU Pods supported only on L4 GPUs. Serverless platforms Cerebrium and Beam describe GPU memory snapshots on their platforms, and Parasail and Photoroom have published measured results. Engine-level support is also emerging: vLLM 0.31.0 includes experimental initialized-engine snapshots for single-GPU TP1 engines (no tensor parallelism across GPUs), and SGLang has an open RFC proposing a similar initial scope. Both are restricted to restoration on the same host and GPU.

Inference: limitations and problems to be solved

A warmed multi-GPU replica cannot simply be copied. Process memory, compiled kernels and CUDA graphs can come back exactly as they were saved, but the connections between GPUs cannot. They must be torn down before the snapshot and rebuilt after restore, and each copy also needs its own identity. The published work stops where those get hard!

© 2026 CLOCKWORK. All rights reserved.

- **Multi-node.** NVIDIA Snapshot's repository states that workloads "must run on a single node" and that multi-node "is on the roadmap." Modal's GPU snapshots target a single GPU. We found no published measurement of an inference replica restored across servers, which is where the largest models run. There are two fundamental problems that need to be solved for multi-node support:
    - **Communication state.** RDMA connections, queues and memory registrations live in the adapter, the kernel and the remote peer, not in the process image. Communicators must be torn down before capture and rebuilt after, with every rank stopped at a consistent point. Modal notes that NCCL programs "frequently deadlock when one peer goes quiet."
    - **CUDA graphs that reference NCCL.** Captured graphs hold memory addresses and references into NCCL. Since NCCL 2.26, NVIDIA documents a strict release order: graphs must be destroyed before communicators. Destroying the graphs means capturing them again, which is a large part of the cost a snapshot tries to avoid.

- **Replica identity and KV-transfer connections.** A snapshot is taken from one replica but restored on many. Each restored replica needs its own identity: engine ID, endpoints and service registrations. Each replica's connections to the rest of the serving stack must also be re-established. In disaggregated serving, where prefill and decode run on different replicas, the connections that transfer KV state between them must also be re-established after restore, with new memory registrations and peer metadata. Many deployments also attach an external KV-cache layer such as LMCache, Mooncake, or SGLang's HiCache. Both kinds of connections register GPU memory with the network adapter and hold handshake state with their peers, and none of it is carried in the process image. NVIDIA's guidance is to turn off transports that CRIU cannot restore.

© 2026 CLOCKWORK. All rights reserved.

- **Capture point.** The workload has to cooperate in its own capture. NVIDIA's workload contract requires the engine to stop in-flight generation, park GPU memory and signal readiness through sentinel files. Modal uses lifecycle hooks, and vLLM's experimental snapshots build the same steps into the engine for a single GPU on the same host. These steps cover the serving engine, but not the components around it, so pausing the engine does not itself make the full stack safe to snapshot. Outstanding transfers and background work must be drained or quiesced. Non-restorable resources must be released before capture and rebuilt afterward, or initialized only after restore, while cache buffers and graph references stay valid. That can require substantive changes to engine or backend initialization, including in components maintained by other projects.
- **Orchestration.** NVIDIA Snapshot handles checkpoint and restore but leaves orchestration (deciding which pods to snapshot, when, and how to restore them, etc.) up to the platform (NVIDIA, Crusoe).

Training: Clockwork provides the industry's only multi-node training snapshots

Although single node snapshots can be applied to training using CRIUgpu (now part of upstream CRIU), distributed training, by definition, requires multi-node snapshots.

In October 2026, Clockwork launched TorchPass Snapshots, the industry's only solution for multi-node training snapshots. In internal testing, Clockwork demonstrated taking a snapshot of a Megatron-LM job on eight H200 nodes, with 72 GiB of GPU memory per rank in under 20 seconds.

The key benefit of training snapshots (vs application checkpoints) is they allow the platform, rather than the model builder, to protect distributed training jobs. Snapshots can be taken periodically as recovery points for unexpected failures. They can also capture a job's state automatically when a node has to be drained, preempted or removed, then restore the job warm, with its runtime already

© 2026 CLOCKWORK. All rights reserved.

initialized. All this can be provided as a platform service without touching the model code.

Conclusion

Snapshots are becoming a fundamental piece of AI infrastructure. Training snapshots save progress; inference snapshots save preparation. Modal showed single-GPU inference snapshots working at production scale, and NVIDIA has extended them to eight GPUs on one node as open source. The open problems are replicas that span servers, communication state, graph references, replica identity and KV-transfer connections.

The hard problems overlap. Training and inference must both handle multi-GPU communication that cannot be copied. Clockwork's TorchPass Snapshots already do this for running multi-node training jobs. That work is the foundation for how we are approaching the harder inference snapshot problems. Stay tuned.

© 2026 CLOCKWORK. All rights reserved.