Solution Brief

KV Cache Offloading on VAST Data: Cutting Agent-Coding TTFT in Half

Backend.AI + VAST Data deliver line-rate KV cache offloading for heavy agent workloads

Inference Is Becoming Stateful

The first generation of LLM serving optimized for stateless prompts: one question in, one answer out. Agent workloads broke that model. A coding agent carries a large working context, the codebase, its tools, its earlier reasoning, and returns to it across many turns. What decides responsiveness is no longer raw token throughput, but how cheaply the engine can re-enter a context it has already processed.
That cost lives in the KV cache, the structure that lets a model reuse prior work instead of recomputing it.

Why GPU-Only KV Cache Falls Short

Agent coding sessions establish a session-specific long prefix and reuse it across multiple turns. Once several such sessions cycle through one inference server, the working set blows past memory.
By the time an agent returns for its next turn, its KV blocks have been evicted by another context, and the engine redoes prefill from scratch. The cache that should accelerate reentry ends up in the most expensive, most contended memory in the system.

The Idea: Move the Cache Off the GPU

If the cache is what makes reentry cheap, the question is where it should live. KV cache offloading takes the blocks the GPU is no longer actively using, writes them to high performance external storage, and pulls them back when the agent returns, instead of recomputing them from scratch. The GPU stops being the only place a context can rest: cold context moves out, hot context stays in, and the engine reloads what it needs on demand.

Why KV Cache Offloading?

Lower TTFT

Repeat-turn cost drops up to 70%

Higher GPU Utilization

Less GPU time spent on avoidable prefill

Line-Rate Data Movement

GPUDirect Storage moves KV blocks between GPU and storage without staging through host RAM

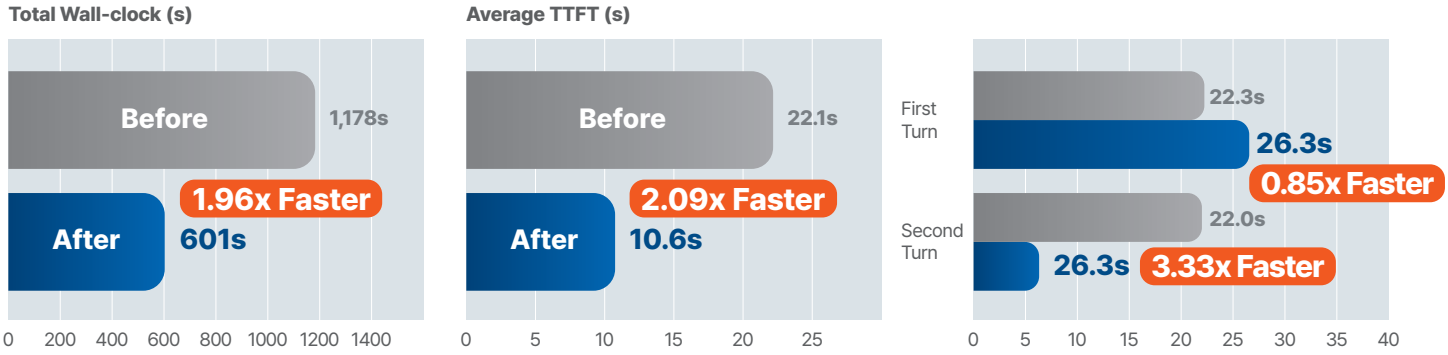
The Backend.AI + VAST Data Solution

Backend.AI is Lablup's enterprise AI infrastructure operating platform for both AI development and production service operations. It is designed to manage heterogeneous AI accelerators, including GPUs, NPUs, and TPUs, within a single operational framework, while also providing structured integration with a wide range of storage systems so that storage capabilities can be used at their full potential. VAST Data storage and the VAST Data AI OS are next-generation all-flash data platforms designed for high-speed AI, machine learning, and high-performance computing workloads, and when combined with Backend.AI, they form the storage layer that enables line-rate KV movement.

In Backend.AI, every compute workload is represented as a session. Each session mounts a VAST KV cache vFolder through Storage Proxy. Inside the session, vLLM and LMCache run together, and LMCache's gds_path points directly to the VAST mount.
Using NVIDIA Magnum IO GPUDirect Storage, Backend.AI and VAST Data move KV blocks directly between GPU memory and storage without passing through host RAM or other intermediate paths.

Benchmark Results

Wall-clock and TTFT both improve by roughly 2x, Competes in the first turn, wins since second turn



- Without offload
- With offload
- Accelerated drainage

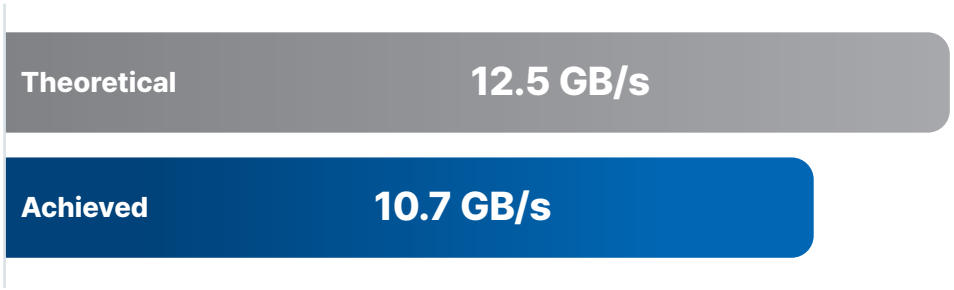
Total wall-clock and average TTFT both improve by roughly 2x, the gain is entirely in getting to the first token. Even so, the average TTFT figure becomes meaningful only when first-visit turns are separated from subsequent reuses against the same context. First turns pay a ~4s offload penalty* because prefill is paired with a storage write. Every later turn reads cached blocks back instead of recomputing, and TTFT drops to roughly a third of baseline. For the user, generation starts after six seconds rather than twenty-two.

* Note: Enabling KV cache offloading introduces several seconds of additional latency on the first turn of a vLLM session. This latency was observed consistently across storage media, indicating that the root cause lies in the LMCache implementation rather than the underlying hardware configuration. Until a fix is available, this behavior should be documented as a known limitation of LMCache.

Line-Rate Throughput for Large-Scale Serving

VAST reads peaked at 10,693 MB/s and 10,441 ops/s, which is line rate against the 100 Gbps fabric, with average round-trip time of 8.265 ms and zero retransmits.
As a result, pulling a long-context KV cache from VAST + Backend.AI is faster than regenerating it on the GPU. Delivering data at the absolute limit of the high speed network ensures that storage performance never degrades, even under intense, multi-tenant workloads.

VAST Reads Throughput



The architecture scales horizontally without bottlenecking at the storage layer, a critical requirement for production-grade LLM serving where hundreds of requests with distinct session contexts arrive in parallel. Whether serving a legal document or a multi-user, multi-turn enterprise-grade chatbot session, the KV cache remains instantly accessible rather than becoming a GPU compute liability.

Benchmark Conditions

The workload runs on 8x H100 GPUs and VAST AI OS v5.4 inside a Backend.AI 26.4 environment with Mistral Medium 3.5 128B: one inference server cycles through ten distinct agent contexts, five turns each, on a 140K-token base. Between visits to any given context, nine other long-context sessions pass through the GPU.

About Backend.AI

Backend.AI is Lablup's enterprise AI infrastructure operating platform, designed for both AI development and service operations. It is built to manage heterogeneous AI accelerators such as GPUs, NPUs, and TPUs within a unified operating framework, while also providing structured integration with a wide range of storage systems so that their capabilities can be used at full performance. Backend.AI also aims to provide a general-purpose execution environment that is not tied to any specific framework, with support for languages such as Python, R, C/C++, Java, and Rust, as well as AI libraries such as PyTorch, Megatron-LM, and vLLM. It is designed to deliver a consistent experience across on-premises, cloud, and hybrid environments.

Lablup has been working on AI infrastructure orchestration since 2015, well before the industry began paying broad attention to the problem. With years of experience simplifying the software layer for AI execution, Lablup provides a foundation for running AI workloads reliably across diverse operating environments. Looking to run AI on any accelerator without rebuilding your stack? Contact us at contact@lablup.com

About VAST Data

VAST Data is the AI Operating System company – powering the next generation of intelligent systems with a unified software infrastructure stack that was purpose-built to unlock the full potential of AI. The VAST AI OS consolidates foundational data and compute services and agentic execution into one scalable platform, enabling organizations to deploy and facilitate communication between AI agents, reason over real-time data, and automate complex workflows at global scale.

Built on VAST's breakthrough DASE architecture – the world's first true parallel distributed system architecture that eliminates tradeoffs between performance, scale, simplicity, and resilience – VAST has transformed its modern infrastructure into a global fabric for reasoning AI.
Exploring ways to accelerate inference with KV Cache? Contact a VAST Solution Engineer at hello@vastdata.com