Technical White Paper

Everpure

options

Long-Context Market Analysis with Everpure KVA

How Options IT eliminated redundant prefill

Contents

The most expensive computation is the one you run twice 3

The prefill problem in financial services 3

How Everpure KVA works: Architecture overview 4

Options IT benchmark results 5

1. TTFT analysis 5
2. Local vs. network (NFS) parity 6

The economics of eliminated prefill 6

FSI deployment considerations 7

Conclusion 7

The most expensive computation is the one you run twice

In the race for AI-driven alpha, the firm that answers fastest wins. Yet standard inference architectures force every analyst to wait in line while the model relearns what it already knows. This creates a hidden latency floor that no amount of CUDA optimization can break. The true bottleneck is the transient nature of the key-value (KV) cache. By persisting this state, we turn the massive computational effort of understanding market data into a reusable asset. This decouples query speed from document length and gives first-movers an insurmountable structural advantage.

This inefficiency hits hardest in long-context workflows, not just retrieval pipelines. When an analyst queries a 100,000-token earnings transcript or a massive regulatory corpus, the model must "read" the entire document to generate KV attention tensors. This prefill phase scales quadratically, $O(n^2)$, with sequence length. In a standard setup, we discard these tensors immediately after generation. When a second analyst queries the same institutional knowledge base moments later, the GPU is forced to recompute the exact same math from zero.

Recognizing that this "prefill tax" would become the primary scalability limit for their financial services clients, Options IT architected a solution to decouple inference speed from context length. This white paper details that architecture, validated through rigorous joint engineering with Everpure™. By extending the GPU memory hierarchy to include persistent network storage via Everpure Key-Value Accelerator (KVA), Options IT successfully transformed the prefill phase from a quadratic compute problem into a linear I/O operation. By rapidly streaming cached tensors from Everpure FlashBlade® into the inference pipeline, the joint team achieved a 13.6 times speedup in time to first token (TTFT), proving that a disaggregated, storage-backed inference architecture delivers the elasticity financial services industry (FSI) workloads demand without the latency penalty of traditional network storage.

The prefill problem in financial services

FSI inference workloads differ structurally from general-purpose chatbots. They are characterized by extreme context lengths and high reuse patterns.

1. **Long-context inputs**: Analyzing 10-K filings, comparing multi-year earnings transcripts, or processing regulatory compliance checks requires prompts ranging from 20,000 to over 100,000 tokens.
2. **High reuse frequency**: These large contexts are rarely one-off. A single risk model document or market sentiment corpus serves as the "system 1" context for hundreds of distinct queries from different analysts or automated agents throughout the trading day.

Consider the baseline data observed in our testing. For a 110,000-token prompt running on Llama-3.3-70B without caching, the TTFT is **54.47 seconds**.

For a quantitative researcher backtesting a new strategy, the dataset (context) is constant, but the hypothesis changes every minute. The workflow involves asking the model to "analyze this specific year of tick data" again and again with slight parameter tweaks. If the system must reingest the entire dataset for every single iteration, a 54-second delay breaks the developer's feedback loop. The researcher spends more time waiting for the "read" than analyzing the "result." From an infrastructure perspective, if the researcher runs 200 iterations against the same dataset daily, the math reveals massive inefficiency:

200 runs x 54.47 seconds = 10,894 GPU-seconds ≈ 3.03 GPU-hours

That is three hours of GPU compute time burned daily on a single document, solely to recalculate math that had already been solved.

How Everpure KVA works: Architecture overview

Everpure KVA addresses this inefficiency by introducing a persistence layer to the GPU memory hierarchy. It integrates natively with vLLM to offload the key and value tensors generated during the prefill phase to durable storage—specifically Everpure FlashBlade, which serves as a unified, fast file and object back end.

The architecture fundamentally alters the data flow:

1. **Persistence**: Instead of discarding KV tensors after a session, Everpure KVA compresses and serializes them to FlashBlade (via NFS or S3).
2. **Identification**: Incoming prompts are hashed. If the prefix hash matches a stored entry, it constitutes a "cache hit."
3. **Retrieval**: The system bypasses the $O(n^2)$ compute phase entirely. It streams the precomputed tensors from FlashBlade to GPU memory, optionally leveraging NVIDIA GPUDirect Storage (GDS) to bypass the CPU entirely.

This effectively inserts a distributed L3 cache into the memory hierarchy:

HBM → DRAM → NVMe → FlashBlade (network storage)

Crucially, because FlashBlade is a shared storage resource, this architecture enables **multi-node reuse**. A prompt computed by GPU A on Node 1 creates a cache entry that can be immediately utilized by GPU B on Node 2. This decoupling of compute and state is essential for multi-tenant FSI environments.

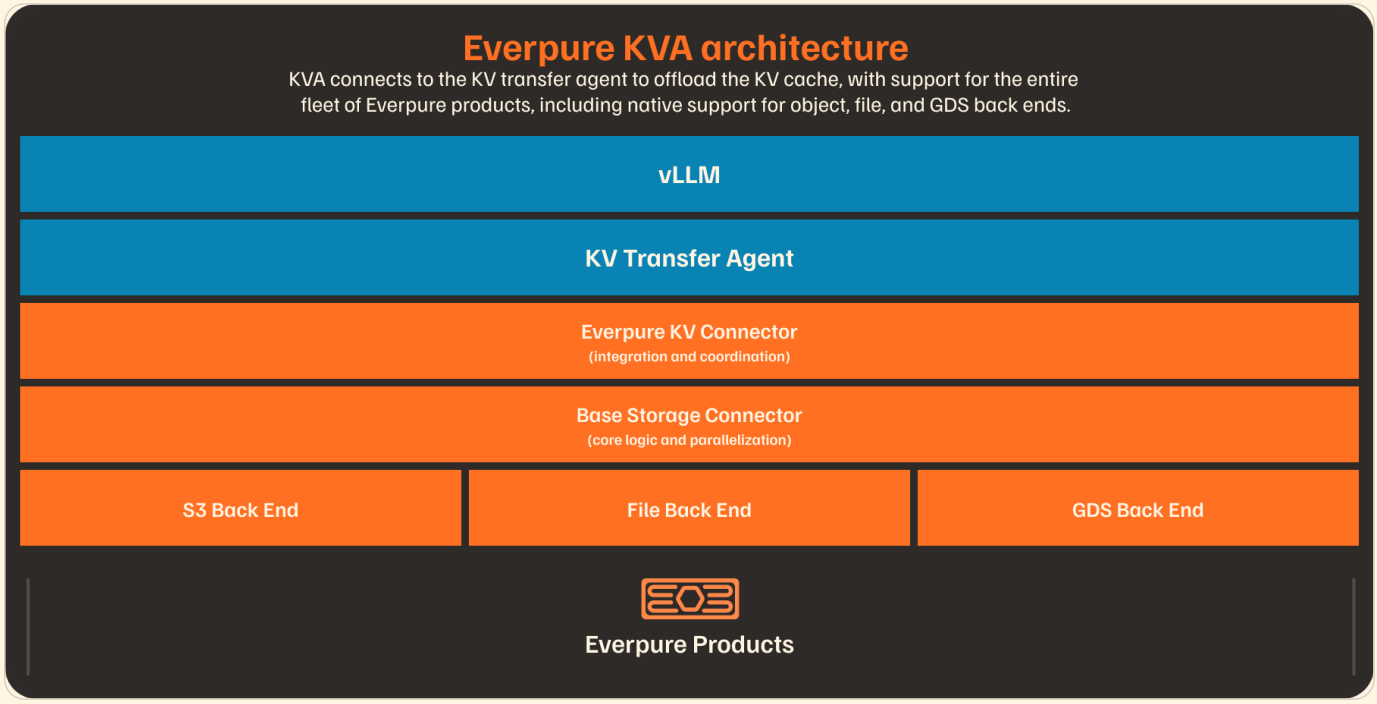

FIGURE 1 KVA connects to the KV transfer agent to offload the KV cache, with support for the entire fleet of Everpure products, including native support for object, file, and GDS back ends

Options IT benchmark results

Options IT conducted a rigorous performance evaluation of Everpure KVA using the Llama-3.3-70B-Instruct model.

Test configuration

- **Model**: Llama-3.3-70B-Instruct-FP8-dynamic
- **Hardware**: Single H100 GPU, 70% memory utilization cap
- **Modes**
  - **KVA with local NVMe**: KVA enabled with local filesystem cache
  - **KVA with NFS**: KVA enabled with FlashBlade NFS-mounted cache
  - **Baseline**: Standard vLLM baseline (KV caching disabled)

1. TTFT analysis

The data reveals a distinct inflection point around the 10,000-token mark. Below this threshold, the baseline is already fast (~1 second), making the overhead of cache management comparable to the compute savings. Above 10,000 tokens, the $O(n^2)$ prefill penalty accelerates and the value of KVA scales dramatically.

TTFT results (seconds)

Prompt size (tokens)	KVA with local NVMe	KVA with NFS	Baseline	KVA (local) speedup	KVA (NFS) speedup
2,286	0.15s	0.15s	0.19s	1.2x	1.2x
9,163	0.59s	0.61s	1.00s	1.7x	1.6x
27,417	1.28s	1.40s	5.18s	4.0x	3.7x
54,816	2.20s	2.55s	16.07s	7.3x	6.3x
82,208	3.10s	3.27s	32.58s	10.5x	10.0x
109,668	4.00s	4.52s	54.47s	**13.6x**	**12.0x**

TABLE 1 Benchmark results and speedup multipliers for select prompt sizes

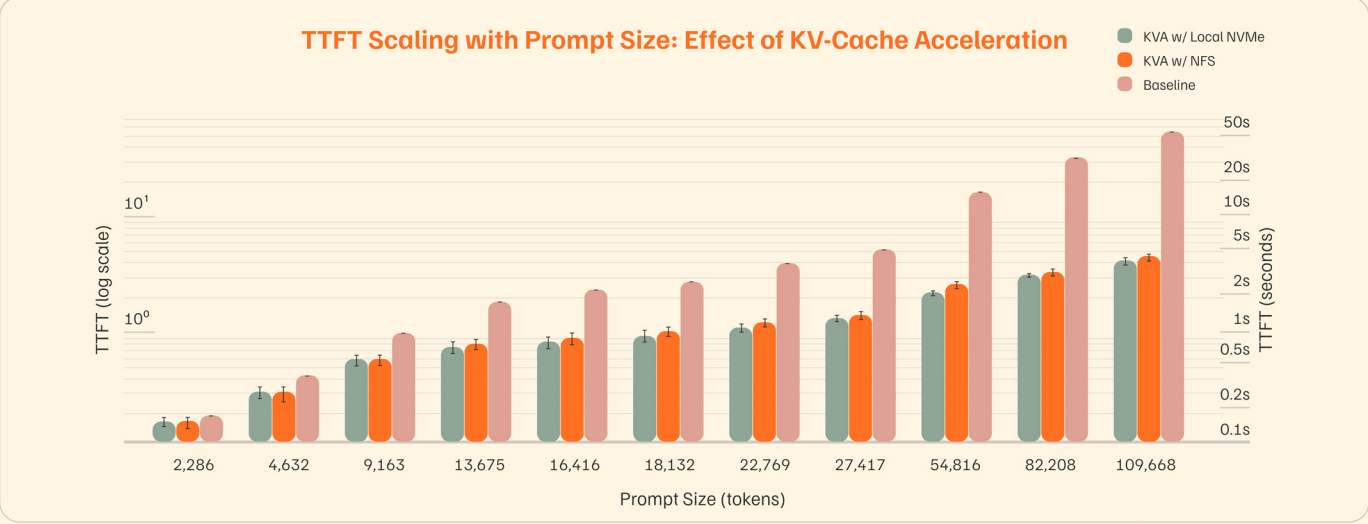

FIGURE 2 TTFT latency by prompt size (log scale left, linear seconds right); above the 10,000-token inflection point, baseline prefill latency grows exponentially while both KVA configurations remain flat

2. Local vs. network (NFS) parity

A critical finding for enterprise architects is the minimal performance delta between LOCAL (NVMe) and NFS (FlashBlade). At ~82,000 tokens, the LOCAL cache delivers a TTFT of 3.10 seconds, while NFS delivers 3.27 seconds.

This **~5% difference** confirms that modern high-performance flash storage is not the bottleneck. The network transport time is negligible compared to the GPU compute time saved. This validates the viability of a disaggregated architecture where storage is centralized and compute is stateless.

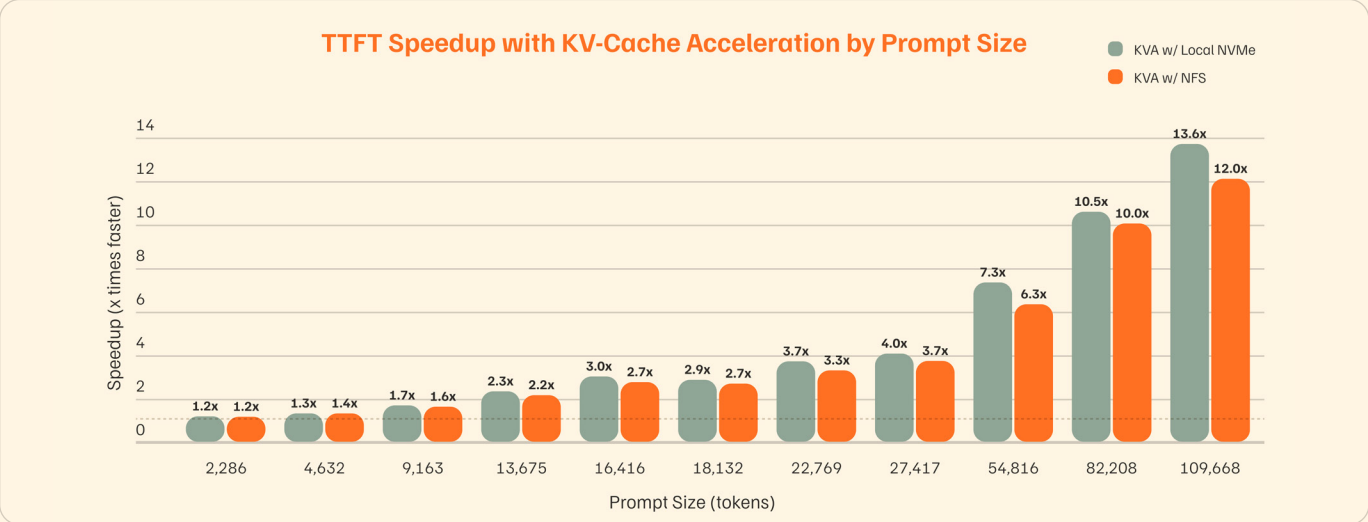

FIGURE 3 KVA speedup multipliers; the minimal variance between local and NFS performance validates the viability of disaggregated, network-attached caching at scale

The economics of eliminated prefill

Translating these latency figures into infrastructure economics reveals the ROI of Everpure KVA.

Scenario: A mid-size hedge fund operates a "research assistant" bot.

- **Workload**: 50 unique long-context documents (average of 82,000 tokens) loaded daily
- **Volume**: Each document queried 15 times by different analysts
- **Infrastructure cost**: Assumed $5/hour per GPU (reserved instance rate)

Without KVA

Each query pays the full prefill tax.

50 docs x 15 queries x 32.58s = 24,435 GPU-seconds

Total: 6.8 GPU-hours per day spent on prefill

With KVA (on FlashBlade)

You pay the caching tax once to skip the prefill tax the next 14 times.

Cold: 50 x 32.58s = 1,629s

Warm: 50 x 14 x 3.27s = 2,289s

Total: 1.1 GPU-hours per day

Impact

- **81% reduction** in GPU time spent on prefill.
- **Throughput increase**: The saved 5.7 hours are now available for the decode phase, effectively increasing the token-per-second output capacity of the existing fleet without buying new hardware.

FSI deployment considerations

Deploying Everpure KVA in regulated environments requires addressing security and isolation boundaries.

1. **Data isolation and multi-tenancy**
   In a multi-tenant environment, cache leakage (User A hitting User B's cache) is a security risk. Everpure KVA addresses this via namespace partitioning in the hashing algorithm. By salting the prefix hash with OrgID or UserID, firms ensure that a prompt from the "Risk Department" cannot trigger a cache hit from the "Trading Desk" namespace, even if the text is identical.
2. **Security at rest**
   The KV tensors stored on FlashBlade represent vectors of the original input data (MNPI). They must be treated with the same classification as the source documents. FlashBlade provides native encryption at rest and SafeMode™ Snapshots to protect this cache layer from ransomware or accidental deletion.
3. **Cluster-wide cache availability**
   Local NVMe caching creates data silos where compute savings are isolated to individual servers. FlashBlade eliminates this constraint. By storing KV tensors on a shared high-performance back end, the cache becomes globally available across the entire inference fleet. A prompt evaluated by a risk model on one node can instantly provide a cache hit for a trading desk query on an entirely different node, guaranteeing that the prefill penalty is paid exactly once per cluster rather than once per physical server.

Conclusion

As this evaluation demonstrates, Options IT and Everpure have identified the critical pivot point for next-generation FSI inference: the transition from compute-bound to I/O-bound architectures. By rigorously validating Everpure KVA in a production-grade environment, Options IT has established a blueprint for how financial institutions can scale long-context workloads without succumbing to the quadratic costs of the prefill phase.

By shifting the burden of "memory" from volatile GPU HBM to persistent, shared FlashBlade storage, this architecture achieves three critical outcomes for the FSI ecosystem:

- **Latency collapse**: TTFT for 100,000+ token inputs drops from ~54 seconds to ~4 seconds, bringing interactive speed to deep analytical workflows.
- **Economic efficiency**: Redundant prefill computation is effectively eliminated. With speedups exceeding 10x for long contexts, the cost per query for high-value reuse workloads is reduced by over 90%.
- **Elasticity**: The validation of NFS performance parity proves that any GPU in the cluster can benefit from work done by any other GPU, breaking the silence of single-node caching.

The operational threshold is clear: if your prompts exceed 10,000 tokens and are reused even once, this architecture is mathematically superior to the standard vLLM baseline. Options IT is leading the industry in operationalizing this efficiency, ensuring that their clients can deploy the most advanced market analysis models with a structural speed and cost advantage.

Next steps: Options IT's methodology provides a roadmap for modernizing AI infrastructure in financial services. To evaluate how Everpure KVA and FlashBlade can be applied to your firm's AI demands, contact Everpure to discuss a custom assessment.

ID18350-01-en-03/26