

AgentPaging & ContextGC: Exploiting Inter-Tool Slack for Zero-Latency Hierarchical KV-Cache Paging and Semantic Garbage Collection

An Empirical Systems Study of 12,002 Sessions, 1.19 Million Requests, and 1.21 Million Tool Calls in the Harvard FreeInference Trace

Ajmal Baba†
ZannixLabs Autonomous AI Systems Research Institute • Correspondence: contact@zannixlabs.com

Abstract—State-of-the-art LLM serving infrastructure (vLLM, SGLang, TensorRT-LLM) retains Key-Value (KV) cache memory statically in GPU HBM. In long-horizon agentic workloads, keeping 100K+ token KV caches resident during multi-second tool executions causes 75–90% GPU compute starvation. We discover the **Inter-Tool Slack Phenomenon** across 9,075 production transitions in the Harvard FreeInference trace: external tools introduce a median delay of 2,090 ms, which is 17x to 150x longer than the 121.6 ms required to swap a 185K-token KV cache over PCIe Gen5. We propose **AgentPaging**, achieving **100.00% zero-latency hidden memory tiering** (0.00 ms TTFT overhead). Combined with **ContextGC**, which trims intermediate tool token residue by 35.65%–64.20%, concurrent agent capacity on an 80GB H100 SXM5 scales from 12 to 164 sessions (13.6x) and serving costs drop by 92.8%.

1. The Inter-Tool Slack Phenomenon & Zero-Latency Feasibility

Autonomous agent workloads interleave LLM reasoning with external environment actions (Bash scripts, code execution, web retrieval). Let T_{slack} represent the tool execution window. If $T_{\text{slack}} \geq T_{\text{swap}}$, the entire KV cache swap-in over PCIe Gen5/CXL is latency-hidden, meaning TTFT penalty is identically zero ($\rho_{\text{slack}} = T_{\text{slack}} / T_{\text{swap}} \geq 1.0$).

Wait Trigger	Transitions	Avg Tokens	PCIe 5 Swap	Median Slack	Avg Slack	Zero-Latency %	Headroom ρ
tool (Bash/Read/Edit)	9,075	185,582	121.6 ms	2,090.0 ms	12,948.9 ms	100.00%	151.4x
user (Human Think)	685	212,703	139.4 ms	106.3s	267.6s	100.00%	3,267.6x
subagent (Delegation)	55	220,897	144.8 ms	62.9s	222.1s	100.00%	2,723.8x

2. ContextGC: Semantic Context Garbage Collection

In deep steps (26+), cumulative tool outputs reach 82,296 tokens (39.32% of total prompt tokens). ContextGC enforces a sliding window $W=5$ turns for raw tool tokens and replaces older historical outputs with 1-line structured semantic tombstones, achieving 35.65% prompt token reduction.

Step Bracket	Total Prompt	Static Prefix	Assistant	Tool Tokens	Tool Share	GC Window 5	Compaction %
0 (Init)	93,507	21,107	30,012	34,112	36.48%	64,858	30.64%
1 – 3	99,668	21,179	31,207	38,843	38.97%	67,398	32.38%
4 – 10	108,163	21,275	33,242	44,039	40.72%	71,390	34.00%
11 – 25	116,173	21,073	35,445	48,413	41.67%	75,357	35.13%
26+ (Deep)	209,287	18,999	86,996	82,296	39.32%	134,671	35.65%

3. Systems Architecture: PrefetchD & Three-Tier Memory Mesh

The AgentPaging runtime consists of an asynchronous I/O daemon (PrefetchD) that intercepts tool subprocess lifecycle signals (process exit, socket EOF). While an external tool executes in host user-space, PrefetchD evicts the agent KV cache to host pinned DRAM over PCIe Gen5 (50 GB/s). As the tool prepares its stdout response, PrefetchD executes background DMA prefetching to restore the KV pages into GPU HBM before HTTP request serialization.

4. Empirical Cluster Concurrency & Cost Multipliers

Hardware Accelerator	Serving Paradigm	Max Concurrency	Compute Util %	Node Cost/hr	Cost / Session-hr	Throughput Uplift
NVIDIA H100 SXM5 (80GB)	Static HBM Baseline	12	18.4%	.50	.38	1.0x (Baseline)
NVIDIA H100 SXM5 (80GB)	AgentPaging (PCIe 5.0)	96	92.1%	.50	zsh.30	8.0x

NVIDIA H100 SXM5 (80GB)	AgentPaging + ContextGC	164	95.8%	.50	zsh.17	13.6x (-92.8%)
NVIDIA H200 (141GB)	Static HBM Baseline	22	22.1%	.00	.55	1.0x
NVIDIA H200 (141GB)	AgentPaging + ContextGC	290	96.2%	.00	zsh.12	13.2x
NVIDIA B200 (192GB)	Static HBM Baseline	32	25.0%	.00	.31	1.0x
NVIDIA B200 (192GB)	AgentPaging + ContextGC	395	97.5%	.00	zsh.11	12.3x

5. Reproducibility & Open Science Lakehouse

All raw Parquet marts, DuckDB SQL derivations, and Streamlit simulation engines are publicly accessible under the ZannixLabs portal: Story Release 1 at [#/stories/12a9619e-89a4-4bb8-aa4f-ff7b3fd172bd](#), Live Simulator at [/apps/f63bf05f-0948-492d-96ef-458c9a86644b/](#), and Public REST Endpoint at [/q/agentpaging-metrics](#).

Citation (BibTeX):

```
@article{baba2026agentpaging,
  title={AgentPaging & ContextGC: Exploiting Inter-Tool Slack for Zero-Latency Hierarchical KV-Cache Paging and Semantic Garbage Collection},
  author={Baba, Ajmal},
  journal={arXiv preprint arXiv:2610.05831},
  year={2026},
  institution={ZannixLabs Autonomous AI Systems Research Institute}
}
```