WHITEPAPER | Q4'25

Driving Dramatic Performance Gains for SideFX Houdini with MEXT Predictive Memory™

An Evaluation of MEXT AI-Powered Predictive Memory™ Technology on AMD Ryzen™ Threadripper™ PRO Workstations

---

Executive Summary

This whitepaper highlights MEXT's transformative performance impact on Houdini, the industry-standard application for 3D simulation. Building rich 3D environments, layering complex animations, and interactively previewing high-quality renders with Houdini all present significant workstation memory demands—ranging from around 16GB to even TBs of memory.

In testing a representative Houdini workload, the job completion time on a 64GB all-DRAM system was 2,000 seconds. We then ran the workload on a 32GB system, with and without MEXT, and then a 16GB system, with and without MEXT. For the 32GB comparison, MEXT drove a 1.3X performance gain. Even more dramatically, for the 16GB comparison, MEXT drove a 4.4X improvement.

In summary, MEXT yields radical performance improvements for Houdini—empowering creators to handle larger jobs and render more, faster. While this particular test case was on a moderately-sized baseline, the observed results would scale to larger-memory situations, as well.

Introduction

Modern animation and visual effects workflows built on SideFX's Houdini often push the limits of creative scope and technical complexity. From procedural modeling and rigging to character animation, shading, and rendering, Houdini enables artists to construct entire worlds with remarkable flexibility. But this versatility comes at a cost: large asset libraries, intricate node networks, and data-intensive simulations all place heavy demands on compute and—especially—memory. As productions grow in scale, depending solely on DRAM becomes increasingly unsustainable, constrained by high costs, limited capacity, and scaling barriers. Addressing these challenges requires a smarter approach to memory management.

MEXT | SideFX

Hardware Configuration

64GB All-DRAM System

- AMD Ryzen™ Threadripper™ PRO 5995WX, 64-core, 128-Thread 4.5 GHz Processor
- 1TB PCIe Gen 4x4, Gen 5x2 M.2 2280 NVMe Internal SSD
- 64GB Total Memory (DDR4 RAM)

32GB All-DRAM System

- AMD Ryzen™ Threadripper™ PRO 5995WX, 64-core, 128-Thread 4.5 GHz Processor
- 1TB PCIe Gen 4x4, Gen 5x2 M.2 2280 NVMe Internal SSD
- 32GB DDR4 RAM

32GB DRAM + MEXT System

- AMD Ryzen™ Threadripper™ PRO 5995WX, 64-core, 128-Thread 4.5 GHz Processor
- 1TB PCIe Gen 4x4, Gen 5x2 M.2 2280 NVMe Internal SSD
- 32GB DDR4 RAM
- 32GB of MEXT Memory™ (via MEXT Predictive Memory™ software enabling NVMe drive to appear as memory)

16GB All-DRAM System

- AMD Ryzen™ Threadripper™ PRO 5995WX, 64-core, 128-Thread 4.5 GHz Processor
- 1TB PCIe Gen 4x4, Gen 5x2 M.2 2280 NVMe Internal SSD
- 16GB DDR4 RAM

16GB DRAM+MEXT System

- AMD Ryzen™ Threadripper™ PRO 5995WX, 64-core, 128-Thread 4.5 GHz Processor
- 1TB PCIe Gen 4x4, Gen 5x2 M.2 2280 NVMe Internal SSD
- 16GB DDR4 RAM
- 48GB of MEXT Memory™ (via MEXT Predictive Memory™ software enabling NVMe drive to appear as memory)

Software Stack

- SideFX Houdini 21.0.440
- Rocky Linux 9.6
- Linux Kernel: 5.14.0-570.39.1.el9_6.x86_64

- MEXT Predictive Memory™ software
- MEXT View™ observability software leveraging Open Telemetry (OTel) and Grafana

---

MEXT AI-Powered Predictive Memory™

How It Works

System memory (DRAM) is one of the costliest components involved in modern computing. However, across many business environments, its utilization regularly drops to 50% or below (demonstrated by various studies from leading cloud service providers and hyperscalers).

MEXT's software solution solves this utilization issue by 1) continuously monitoring which memory pages in RAM actively being utilized, or "hot", and which have gone "cold", 2) offloading the cold memory from RAM to flash, and 3) leveraging AI to mitigate the effects of flash latency and keep the system performant (via the MEXT Predictive Memory™ Engine).

The MEXT AI-Powered Predictive Memory™ Engine continually predicts which offloaded pages might be requested by the application soon (in other words, which pages are likely to soon go from cold to hot), and transparently moves them back into DRAM before the requests are even made. As a result, the application stays performant because from its perspective, the relevant memory pages are always already resident in DRAM.

Value

This empowers customers to run applications performantly within a much smaller DRAM footprint, with far better utilization of that DRAM footprint—yielding substantially lower costs.

In other cases, however, there may be customers whose applications are running out of memory and suffering performance issues as a result. In the past, they would have had to either procure a larger system or shard their data across multiple smaller systems—both of which are complex and expensive paths. With MEXT, customers can cost-effectively expand the effective memory capacity of their system by leveraging flash as memory.

Seamless Implementation

MEXT is a patent-pending, software-only solution that works with any configuration: on-premises or cloud, with any processor, in virtualized / bare-metal / containerized environments, with no changes to the OS or applications. Installation takes less than 5 minutes.

MEXT Solution Components

The MEXT Predictive Memory™ solution consists of 3 primary components: the MEXT Driver, the MEXT Predictive Memory™ Engine, and the MEXT View™ Observability Platform.

MEXT Driver

The MEXT Driver is a dynamically loadable kernel module (which does not alter the standard Linux kernel) that sends process and memory page telemetry data to the MEXT Predictive Memory™ Engine.

MEXT Predictive Memory™ Engine

The MEXT Predictive Memory™ Engine is a user-space process that feeds predictions of which memory pages should be pushed from flash to RAM—making predictions / inferences in under a fraction of a second. It runs entirely on the local Linux operating system (on a single CPU core) and does not require a GPU.

It was inspired by modern AI techniques based on neural networks. Instead of using these techniques to predict words or natural language patterns (like ChatGPT does), it predicts sequences of future memory page accesses. It consists of a family of models that work together, combining extremely lightweight heuristic predictors with more powerful neural-network models. For any given workload, it automatically adjusts to use the model or group of models that performs best. Continuous observation of which predicted pages were actually used by the application also enables the engine to acquire real-time feedback regarding model accuracy, supporting ongoing adaptation and self-optimization.

MEXT View™ Observability Platform

MEXT also provides a user-space application called MEXT View™ that provides observability / visualization tools to help customers profile their workloads—illustrating how much memory their applications are using at any given time and what portion of this memory is hot / warm / cold. All cold memory pages are good candidates for optimization by MEXT Predictive Memory™ software. MEXT View™ also provides insight into the ongoing prediction accuracy of the MEXT Predictive Memory™ Engine.

Methodology

In evaluating Houdini, we used hscript to render several frames in a splash animation sequence. We rendered the first 10 frames which consumed approximately 68GB of memory apiece, with memory

utilization rising and falling through each frame in the sequence. Throughout this period, workload run time, system-level metrics—CPU load, memory footprint, and disk I/O—were continuously tracked. In parallel, MEXT Predictive Memory™ metrics were monitored, including prediction accuracy, memory temperature, and latency improvements.

Results

Baseline

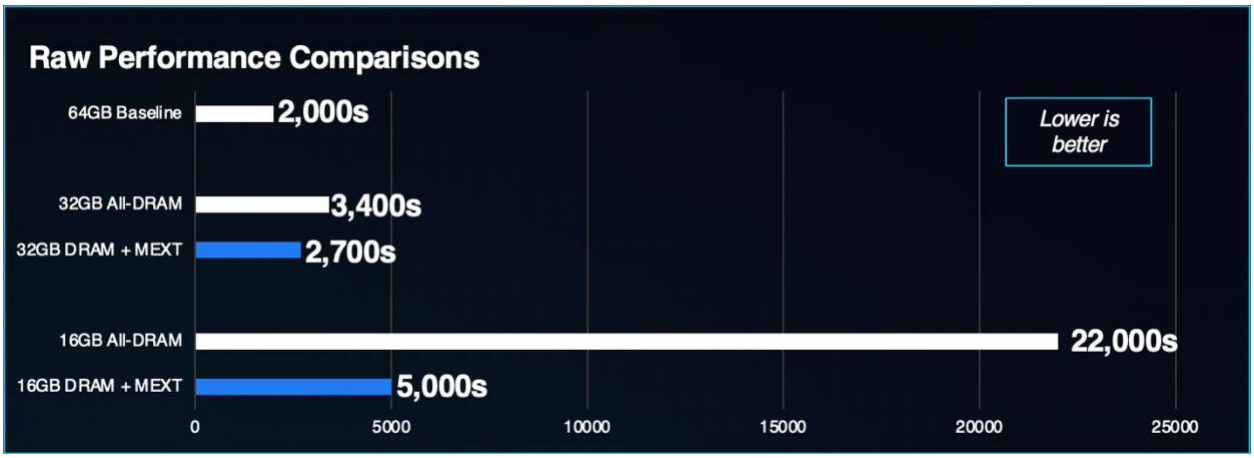
The original configuration (64GB) system actually needed closer to 68GB of memory, necessitating traditional Linux swapping to disk. The overall time it took to complete the job, with swapping enabled, was around 2,000 seconds.

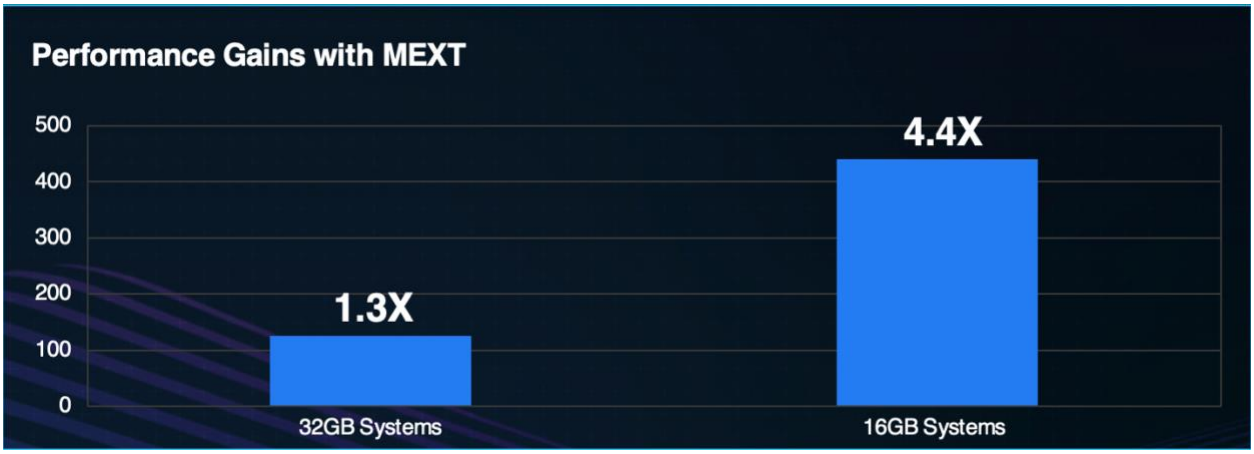
Performance: 32GB Systems

When running the same job on a system with 32GB of DRAM, with swapping enabled, the job completion time increased to around 3,400 seconds (a 1,400-second latency increase vs. the original configuration). Upon adding MEXT to this 32GB system, the job completion time was around 2,700 seconds (a 700-second latency increase vs. the original configuration). This translates to nearly a 1.3X performance improvement with MEXT (3,400 / 2,700 = 1.26).

Performance: 16GB Systems

When running the same job on a system with 16GB of DRAM and swapping enabled, the job completion time increased to around 22,000 seconds (a whopping 20,000-second latency increase vs. the original configuration). The dramatic increase in runtime clearly shows the workload required more memory than the system had available. Upon adding MEXT to this 16GB system and configuring 64GB of MEXT Memory™, the job completion time was only around 5,000 seconds. This translates to a dramatic 4.4X performance improvement with MEXT (22,000 / 5,000 = 4.4).





Conclusion

When running heavy Houdini scenes (such as this example, which required around 68GB of memory), MEXT unlocks performance that simply isn't possible on typical workstation configurations. On a 32GB DRAM + MEXT system, a **1.3X performance gain** was achieved vs. the 32GB all-DRAM system. On a 16GB DRAM + MEXT system, the gain is even more dramatic—**4.4X better performance** vs. the 16GB all-DRAM system.

This means artists and creators don't have to hit the dreaded memory wall that slows down renders or crashes projects. Instead of being forced to upgrade to more expensive / higher-DRAM workstations when their memory needs get too large, Houdini users can keep creating on the systems they already own, with MEXT smoothing out the memory bottlenecks.

The result? More complex scenes, more iterations, and more finished shots—without the frustration of waiting on slowdowns or being blocked by hardware limits. MEXT gives creators the freedom to focus on their art, not their system specs.