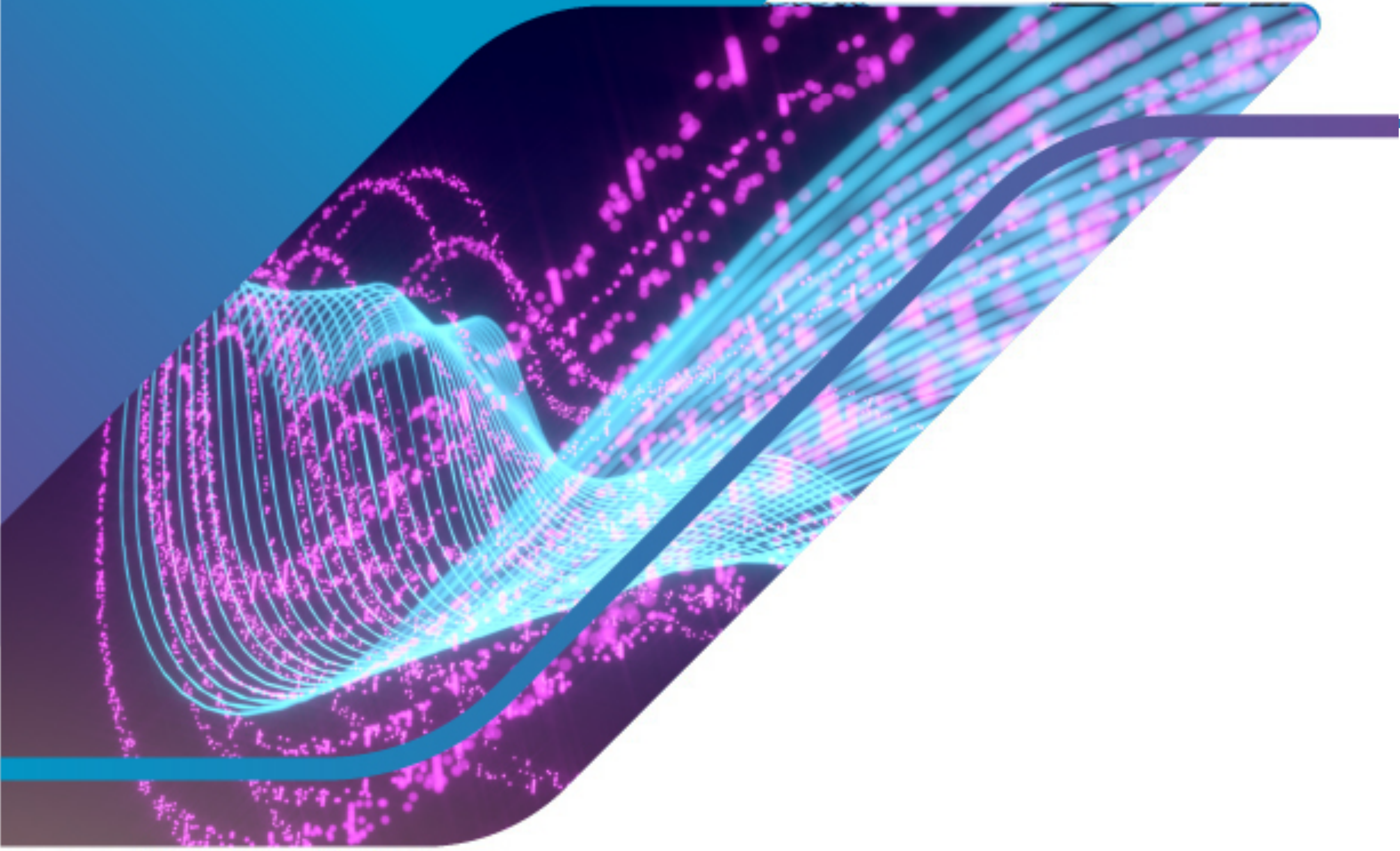
PERFORMANCE STUDY
July 2025



Memory Tiering Performance

VMware Cloud Foundation 9.0

vmware®
by Broadcom

Table of contents

Executive summary

Memory Tiering in VMware Cloud Foundation® 9.0 (VCF 9.0) uses two types of memory devices to increase memory capacity at a reduced cost. Memory Tiering transparently monitors VM memory activity and places frequently accessed data on the fast, more expensive DRAM, while demoting infrequently accessed data to the secondary, slower, cheaper memory tier located in NVMe storage.

This approach optimizes total cost of ownership (TCO) while maintaining comparable performance to systems using only DRAM. Our testing shows that Memory Tiering delivers performance with a loss of less than 10% across various enterprise workloads, as seen in our performance testing with the industry-standard benchmarks VMmark, Login Enterprise, DVD Store, and HammerDB. Memory Tiering also doubles VM density and provides up to a 40% TCO saving.

Introduction

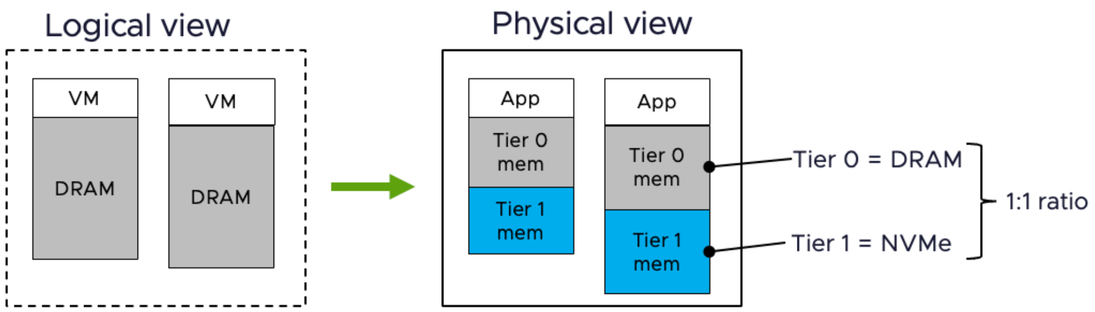
Memory is often the biggest component in the total hardware cost of a server. Modern applications increasingly consume more memory due to the growth in data being collected, make more complex computations, and require real-time operation.

Production deployments tend to not over-commit memory. This is because of unpredictable performance degradation if memory reclamation using ballooning, compression, or swapping happens. These memory management techniques do not have the intelligence of Memory Tiering to make the best decisions about how to manage active memory. In response to ballooning, compression, and/or swapping, admins often decide to provision the total memory that the VM requires instead of over-committing it. This solution works, but it is inefficient because not all of the memory is actively used at the same time. This results in the under-utilization of expensive memory.

In VCF 9.0, Memory Tiering presents a single logical memory space for VMs. Under the hood, however, it manages the Tier 0 (DRAM) and Tier 1 (Memory Tiering) memory types based on VM memory activity. Essentially, it works to keep active “hot” memory on DRAM and inactive “cold” memory on NVMe:

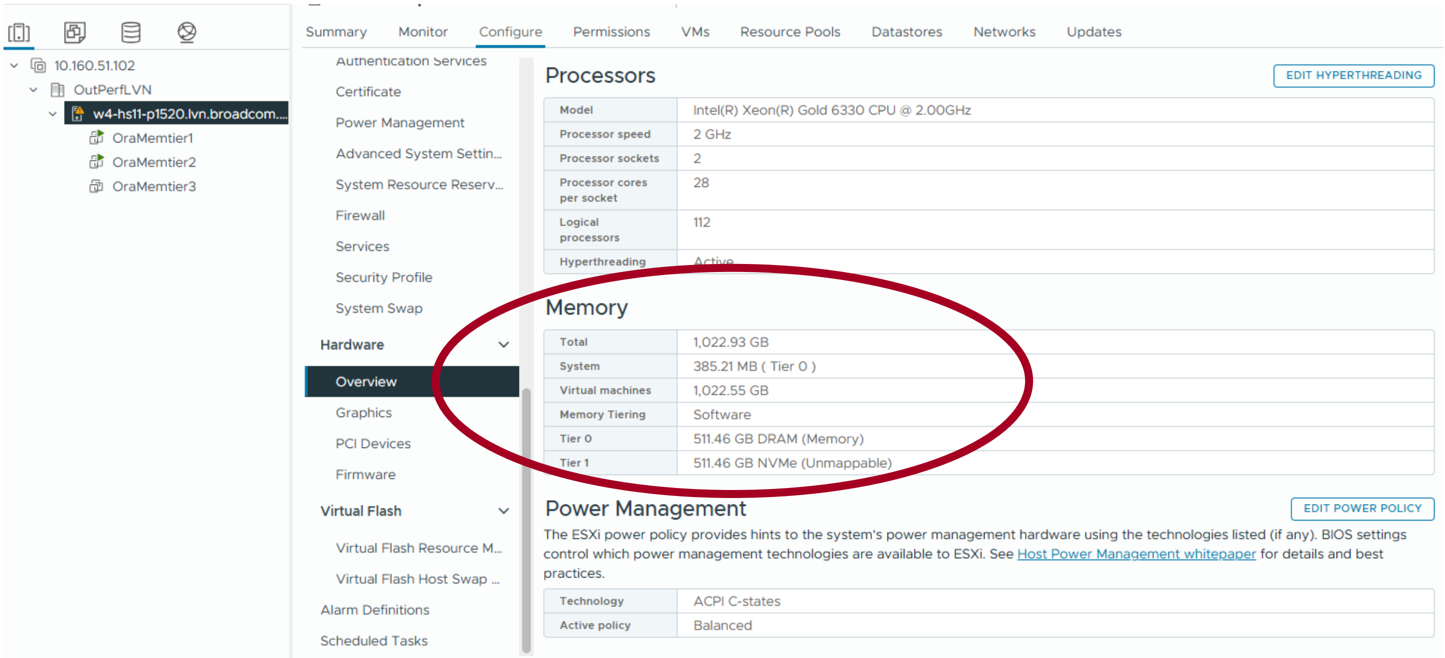
- **Tier 0:** High-speed **DRAM**, which is expensive, very fast system RAM. Memory Tiering retains active ("hot") memory on DRAM.
- **Tier 1: NVMe** devices, which are high-performance solid-state drives (SSDs) that are slower and less expensive than DRAM. Memory Tiering allocates inactive ("cold") memory to NVMe.

Figure 1. Memory tiering is transparent to VMs



After you enable Memory Tiering, you can see its information in the VMware® vCenter® interface: **Configure** tab ➔ **Hardware** ➔ **Overview** ➔ **Memory**. Figure 2 shows the total memory as 1,022.93GB and how much is in each tier: Tier 0 has 511.46GB DRAM and the same for Tier 1 NVMe.

Figure 2. Screenshot of Memory page in vCenter showing total memory at 1,022.93GB with 511.46GB each for Tier 0 DRAM and Tier 1 NVMe.



Processors

Model	Intel(R) Xeon(R) Gold 6330 CPU @ 2.00GHz
Processor speed	2 GHz
Processor sockets	2
Processor cores per socket	28
Logical processors	112
Hyperthreading	Active

Memory

Total	1,022.93 GB
System	385.21 MB (Tier 0)
Virtual machines	1,022.55 GB
Memory Tiering	Software
Tier 0	511.46 GB DRAM (Memory)
Tier 1	511.46 GB NVMe (Unmappable)

Power Management

The ESXi power policy provides hints to the system's power management hardware using the technologies listed (if any). BIOS settings control which power management technologies are available to ESXi. See Host Power Management whitepaper for details and best practices.

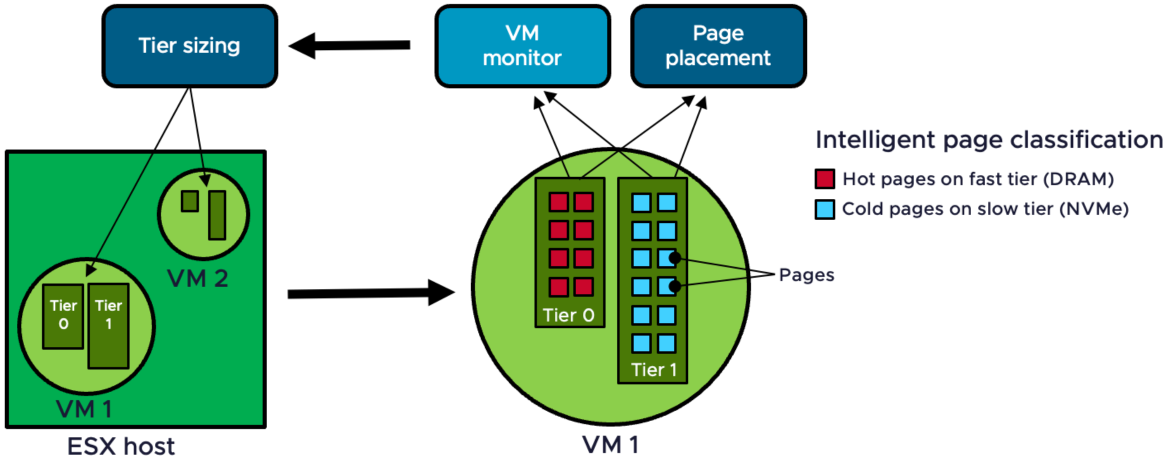
Technology	ACPI C-states
Active policy	Balanced

Memory Tiering architecture

The Memory Tiering system introduces a module that performs memory classification and placement. This module iterates through all guest memory in a periodic fashion, dynamically computing guest activeness, memory tier bandwidth, VM tier quotas, and per-tier page activeness thresholds. The memory scheduler leverages these factors, along with the historical activeness of each guest page, to place the pages on the appropriate memory tiers.

Figure 3 shows how Memory Tiering works in the background. It monitors VM memory access and determines which memory pages are hot and cold in a specific time window. For example, Memory Tiering classifies hot pages as those pages accessed most frequently in the past minute and keeps them in DRAM. It classifies the rest of the pages as cold, sending them to slower NVMe. It continues to monitor and update the hot/cold classification, capturing the changes in the VM's memory activity over time. As the VMs on the system go through workload phase changes (different sections of memory active at different times), Memory Tiering continues to adjust the pages on the tiers and keep the system's DRAM utilization efficient.

Figure 3. Memory tiering intelligently places hot pages in Tier 0 DRAM and cold pages in Tier 1 NVMe


Tier sizing
VM monitor
Page placement
Tier 0
Tier 1
VM 1
VM 2
ESX host
Tier 0
Tier 1
Pages
VM 1
Intelligent page classification
🟥 Hot pages on fast tier (DRAM)
🟦 Cold pages on slow tier (NVMe)

Performance testing

To measure the performance of Memory Tiering, we simulated a production environment virtualized with VCF 9.0 and enabled Memory Tiering. We then ran a variety of workloads on the test system and carefully noted the benchmark results.

Benchmarks used and summary of results

We conducted a wide variety of experiments using popular enterprise-level benchmarks that use and/or stress many components of the system, including the CPU, memory, storage, and network. Perf test: table 1 shows the industry-standard benchmarks used and a summary of the results.

Table 1. Industry-standard benchmarks used and summary of results*

Benchmark	Workload	Results
Login Enterprise	VDI app	2x VM density increase, 0-8% performance loss
VMmark	Enterprise apps	2x VM density increase, 5% performance loss
DVD Store	Oracle Database	2x VM density increase, less than 5% performance loss
HammerDB	SQL Server MySQL	2x VM density increase, 5-10% performance loss (SQL, MySQL)

* All the results in the chart are for 2x VM density. At a lower density of VMs, the performance impact will be lower or negligible.

Memory configuration

The default DRAM:NVMe ratio is 1:1 for VCF 9.0. This means that if you have 1TB of DRAM, you can get an additional 1TB of memory backed by the NVMe device, for a total of about 2TB of system memory (minus some overhead memory) available for the VMs to use. We used this default 1:1 ratio for all tests.

Reporting methodology

We discuss the performance results for various benchmarks and present the following metrics:

1. VM density increase with more memory capacity added by Memory Tiering
2. Performance difference compared to an equivalent system using only DRAM
3. CPU utilization:
    a. How much more CPU can be utilized with an increased memory capacity and higher VM density made possible by Memory Tiering?
    b. Additional CPU overhead added by Memory Tiering

Note: The CPU utilization number plotted in all charts is **% Core Utilization**.

We present two sets of results for each benchmark and/or each configuration in the benchmark.

To accurately depict the performance overhead of Memory Tiering, we left the host large pages disabled in all tests. This is the default setting of an ESX host configured with Memory Tiering.

Active Memory metrics

The **Active Memory** metric is named differently, depending on which tool you use. These counters all indicate the host's active memory usage:

- **esxtop** reports **TCHD** (touched memory) in MB: This appears on the memory screen of esxtop. There is no host-wide TCHD active counter in esxtop. You must sum up the TCHD of all VMs to get the total TCHD for the host.
- **vCenter** reports **Active** in KB: Go to **Monitor ➔ Performance ➔ Overview**. In the **Memory** section, look for the **Active** counter.
- **VCF Operations** reports guest active memory in KB: Go to **Metrics ➔ Memory ➔ Guest Active**.

VDI performance with Login Enterprise

Login Enterprise by Login VSI is industry-standard software for benchmarking VDI capacity and performance. Its virtual user technology simulates real users performing typical tasks while measuring the response times for each interaction. The platform evaluates desktop performance, application performance, and user experience to assess overall VDI responsiveness.

We set up the Login Enterprise benchmark as recommended in its documentation. We used Omnissa Horizon as our virtual desktop infrastructure and installed target VDI VMs on the systems under test (SUTs). We put launcher VMs (that started the workload) and the Login Enterprise virtual appliance (that gathered performance data) on a separate driver system.

The Login Enterprise benchmarking tool includes two preconfigured workloads: the task worker and the knowledge worker. We used the knowledge worker profile because it is the heaviest and most widely used profile in industry benchmarking practices. It has nine different applications, including Microsoft Word, PowerPoint, Excel, Outlook, Edge browser, and video streaming, among others.

The benchmark's main metric is the end-user experience (EUX) score, which measures various actions or timers that simulate typical VDI users working on the system. We used these actions to evaluate application responsiveness, keyboard input processing, CPU-intensive tasks, and latency in storage I/O.

We ran Login Enterprise on a single node without vSAN and on a 3-node vSAN cluster for comparison.

Single node testing

We conducted two experiments, one with inter-VM Pshare disabled, called ModeB provisioning, and one with inter-VM Pshare enabled, called ModeA provisioning. In ModeA provisioning, which uses instant clones, VMs are cloned from and share the memory of a parent VM, typically one per host, when they are first created. In ModeB provisioning, VMs are cloned from a replica VM that is powered off, and they do not share memory with other VMs in the desktop pool. The performance characteristics of both modes are different not just in page sharing behavior, but also in how the Memory Tiering algorithm behaves; consequently, we included testing results for both modes.

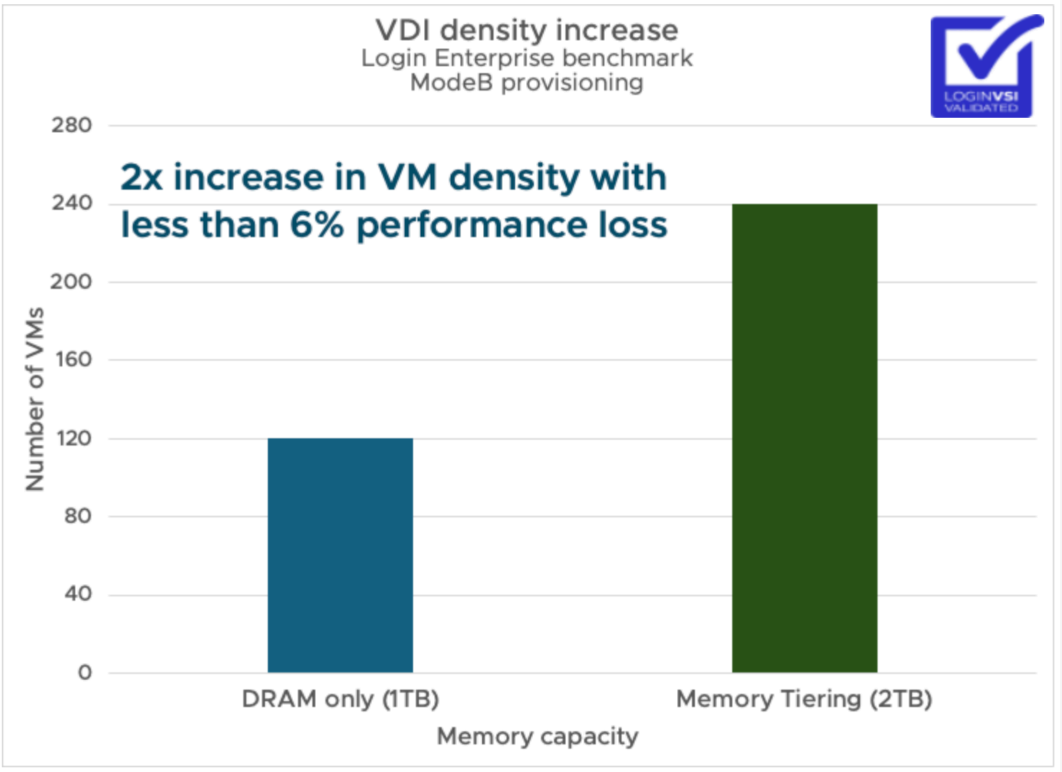
Table 2. Hardware and software used in single node tests

Component	Configuration and version
System under test (SUT)	Dell PowerEdge R760 with a dual-socket Intel Xeon 8480 (56 cores per socket), 1TB and 2TB DRAM
NVMe device	Dell_Ent_NVMe_P5620_MU_1.6TB
SUT storage	P5620 for VDI instant clones
VDI VM configuration	2 vCPU 8GB/6GB Windows 11
Benchmark	Login Enterprise 5.14.7
Benchmark configuration	5 sessions per minute logon with 30-minute test duration
Instant clone provisioning	Omnissa Horizon 8 2312

ModeB performance test results

For ModeB, we configured the VM with 2 vCPUs and 8GB RAM. With 1TB DRAM on the host, this meant we could run roughly 120 VDI sessions (leaving some room for overhead memory). Each session ran in a single VM. By adding the extra 1TB of memory provided by the NVMe device, we could run 120 more sessions for a total of 240 VDI sessions (each running on one of the 240 VMs).

Figure 4. ModeB performance testing showed a 2x increase in VM density with less than 6% performance loss



The Memory Tiering (2TB) configuration has less than a 6% performance loss compared to the DRAM only (1TB) configuration. The EUX score decreased from 8.6 to 8.3 when comparing a higher cost system with 2TB DRAM vs a much lower cost system with 1TB DRAM plus 1TB provided by Memory Tiering. This is a drop of only about 3.5%.

In addition, CPU utilization increased from 63.5% to 74% with Memory Tiering because more data is tiered in and out, which has an associated CPU cost.

Some of the application response times increased as well. For example, Outlook opened in 1.10 seconds in the system with 2TB of Memory Tiering vs 0.94 seconds with 2TB of DRAM—a difference of only 0.16 seconds. Similarly, starting and opening applications like Excel and PowerPoint increased only by about 0.002 seconds.

Figure 5. The EUX score decreased by 3.5% only due to Memory Tiering

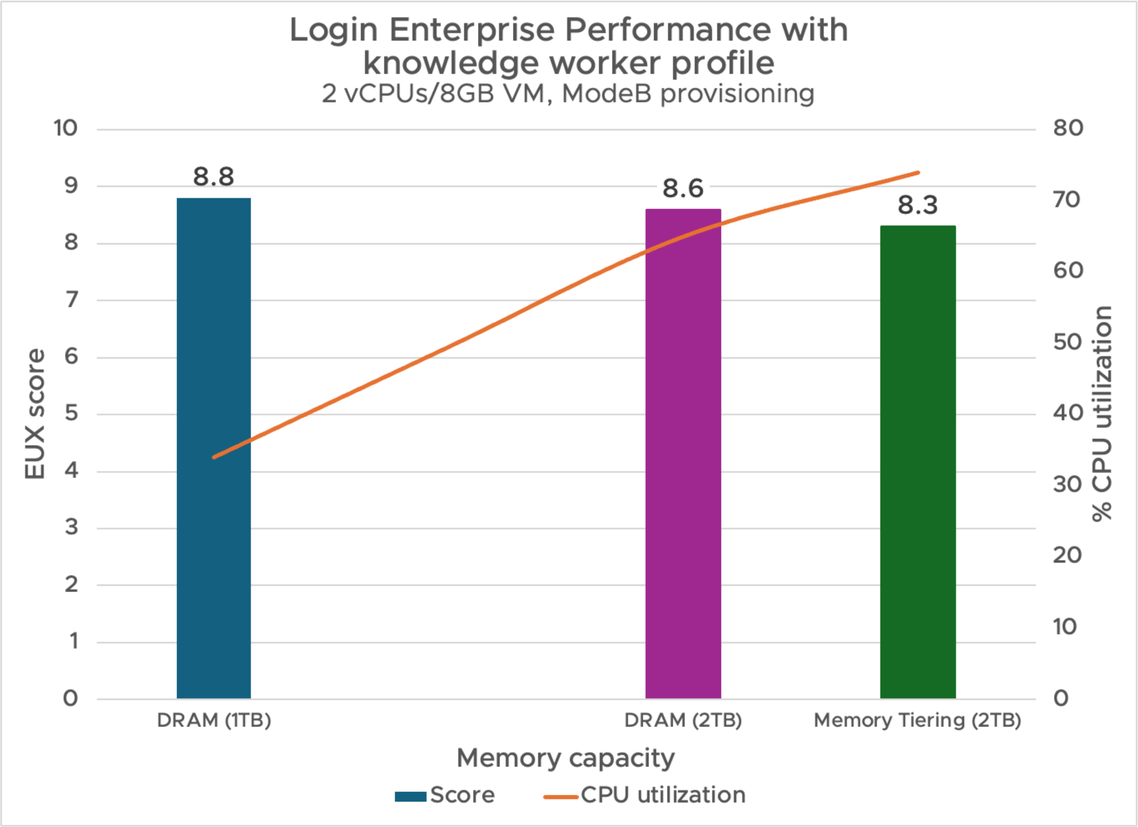
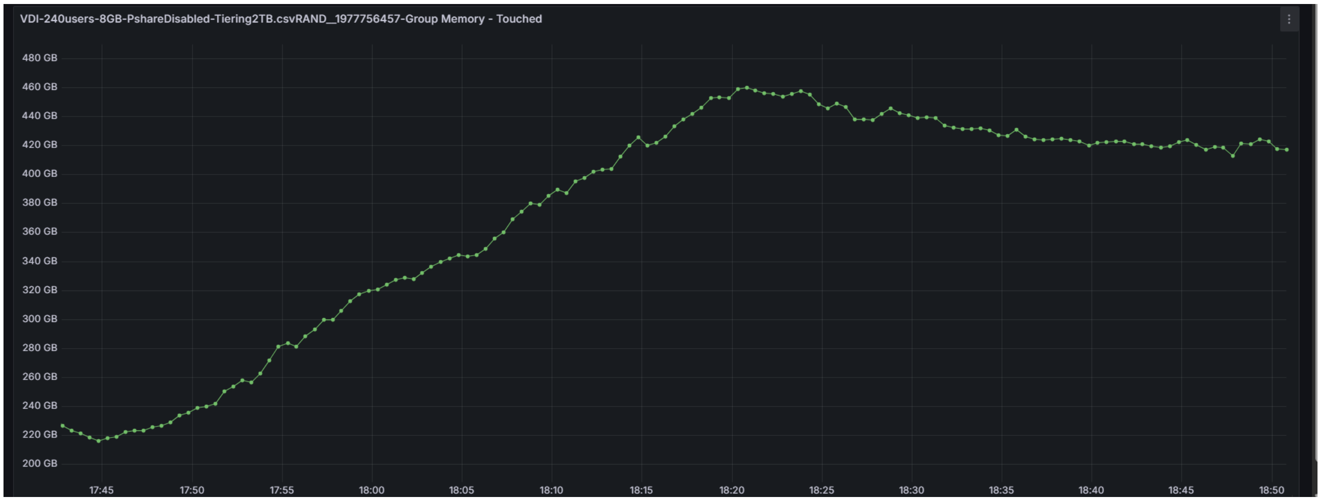


Figure 6 shows the total active memory of the ESX host, which is around 420-450GB. That's about 50% of the DRAM (1TB) capacity.

Figure 6. Total Active Memory of the ESX host is about 420-450GB



The next two charts show the NVMe guest read latency and the NVMe device bandwidth. These are useful in understanding the performance behavior throughout the benchmark, since the NVMe device bandwidth is a proxy for DRAM miss rate and latency is a proxy for the cost of a DRAM miss.

Figure 7. NVMe guest read latency

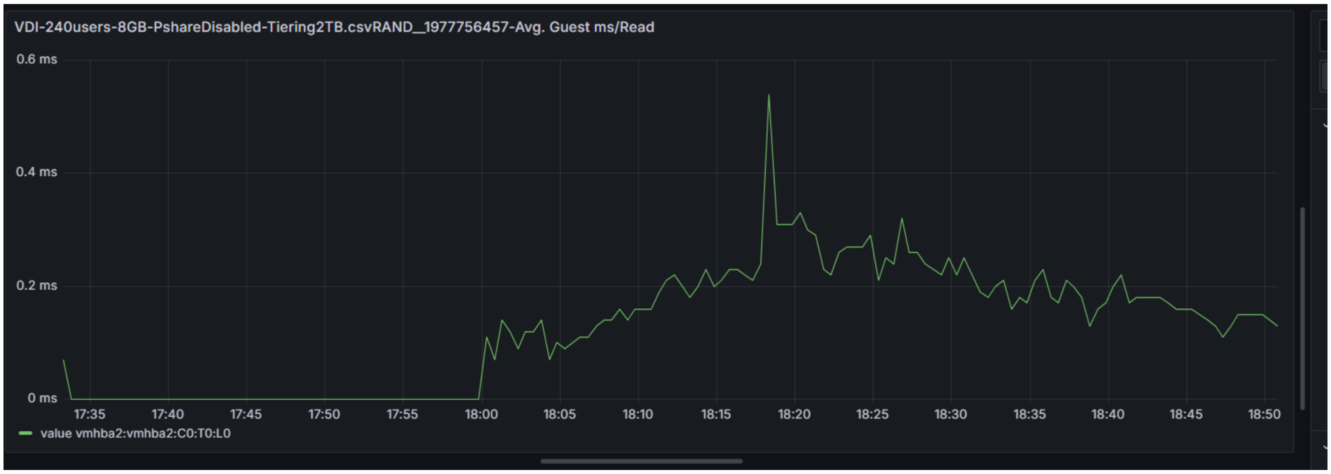


Figure 8. NVMe device read bandwidth

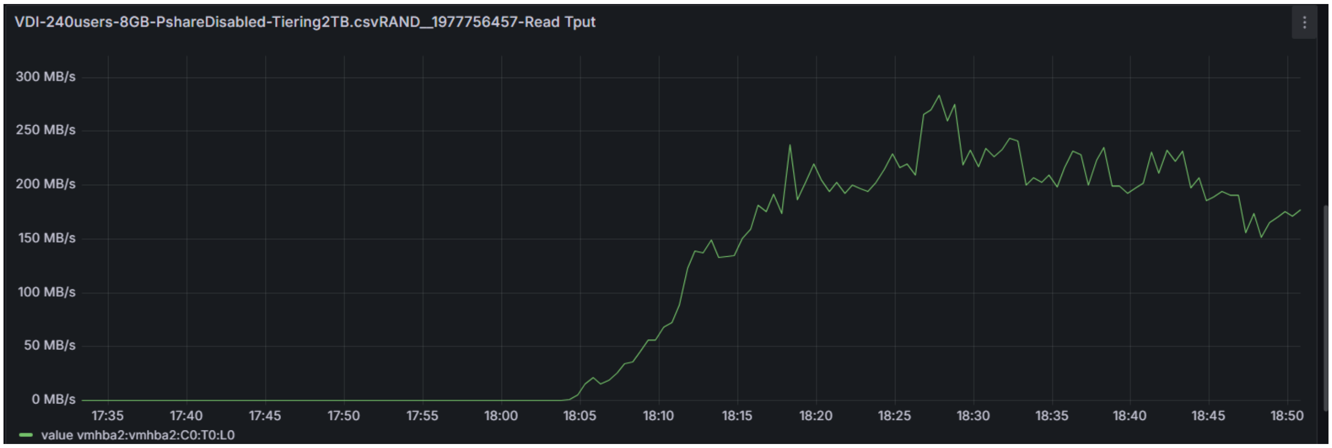


Figure 9 shows the EUX score as the number of VDI sessions increase in the 2TB (1TB DRAM + 1TB NVMe) Memory Tiering case.

- On the left side of the chart, you can see the EUX score drop and then go up a bit.
- On the right side, two interesting lines are CPU score and Generic application score. If you match the trend of these two lines, they correlate very closely to the NVMe device latency and read BW as shown in figures 7 and 8.

After the Memory Tiering latency went above 200 microseconds, these two metrics dropped. When the latency settled at around 200 microseconds, the two metrics started to go up, and so did the EUX score. This shows how the benchmark's EUX score directly reflects the latency behavior of Memory Tiering as the VDI sessions increase, reach a steady state, and then stabilize.

Figure 9. EUX metrics: EUX score and EUX timers

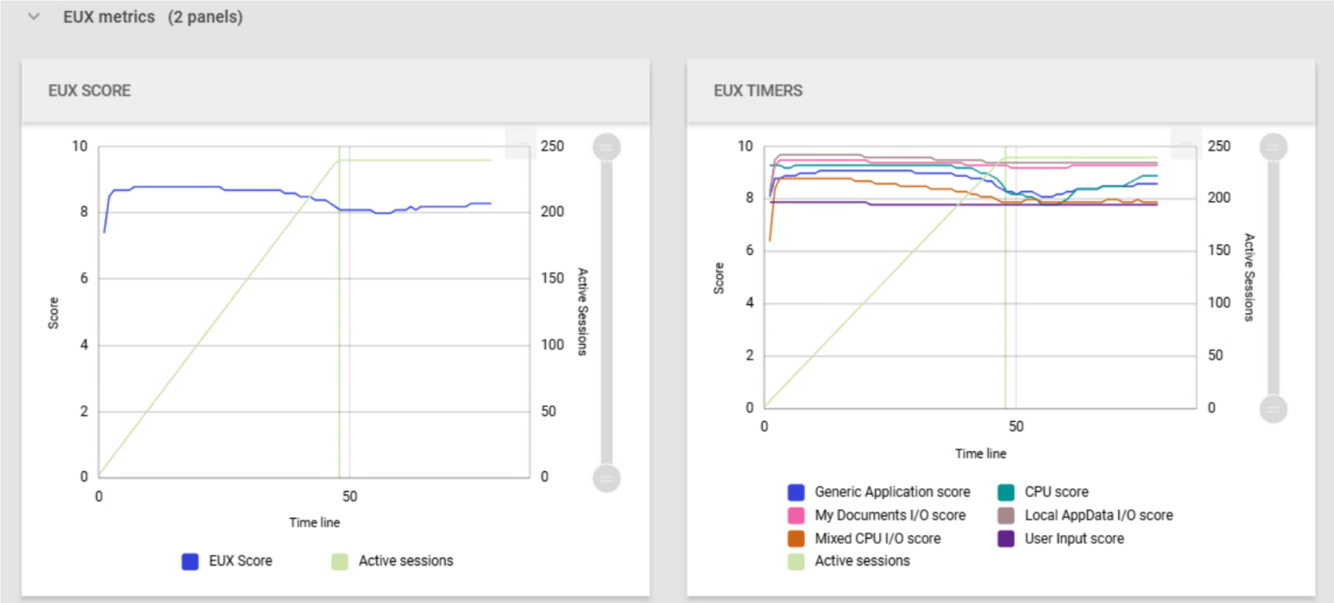
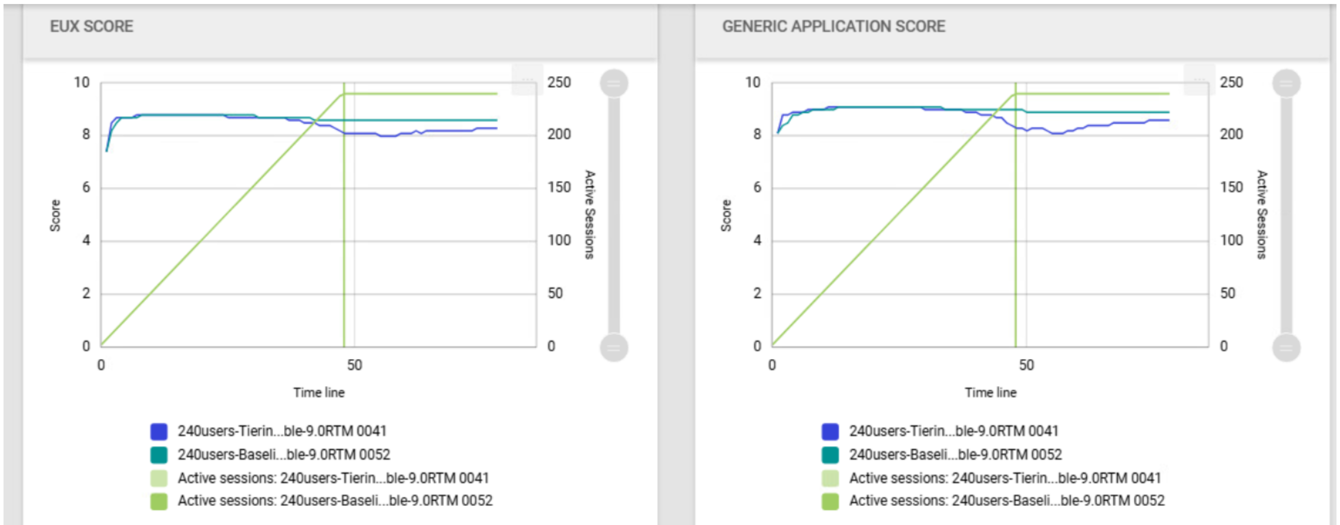


Figure 10 shows how the EUX score and Generic Application score, which has the most weight in EUX score calculation, drops just a little with Memory Tiering enabled vs the baseline DRAM (2TB).

Figure 10. EUX metrics: EUX score and generic application score

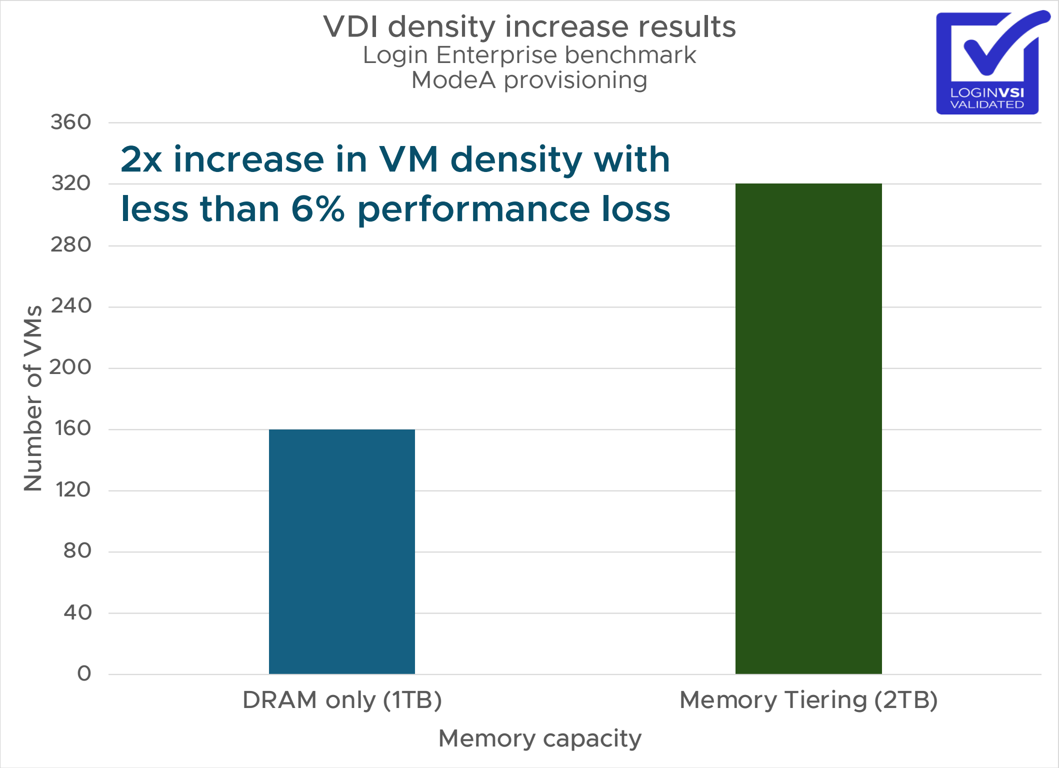


When large pages were disabled in DRAM (2TB), there was no difference in score; however, there was an increase in CPU from 63.5% to 68%. This increase in CPU was not enough to cause any change in the EUX score.

ModeA performance test results

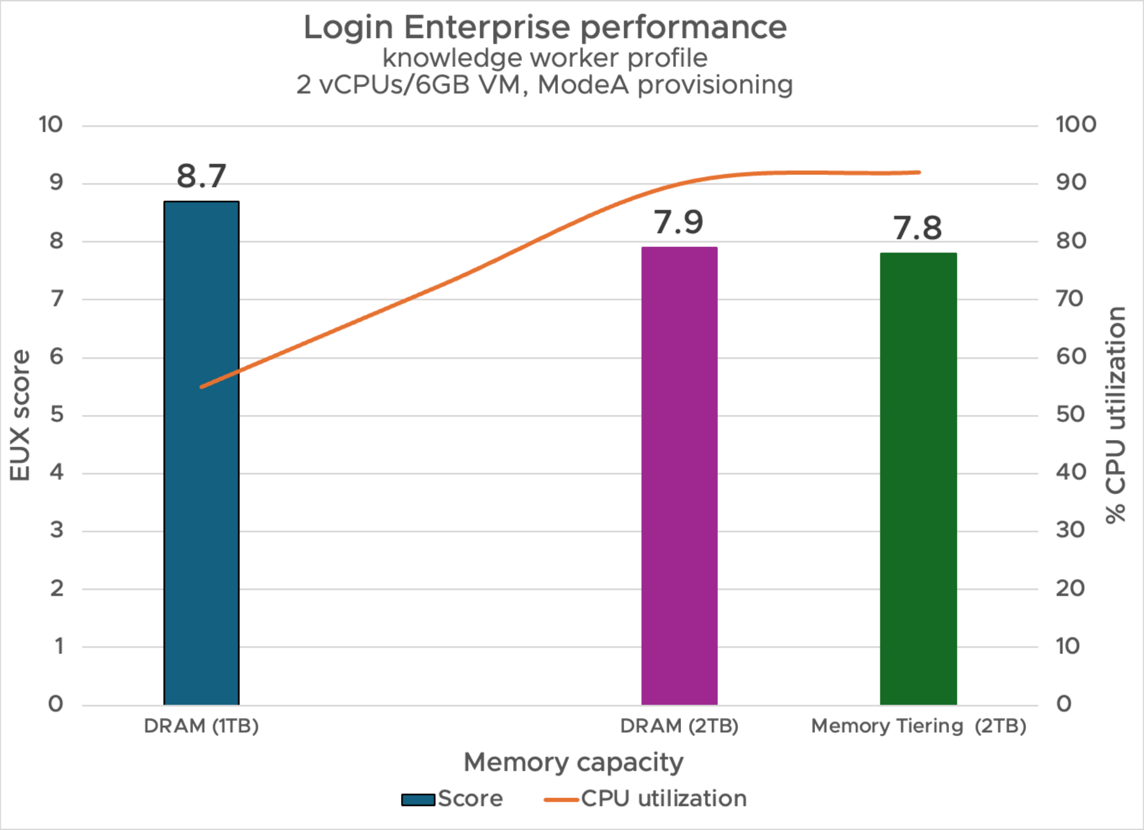
For ModeA, we configured the VM with 2 vCPUs and 6GB RAM because the page sharing benefit gives a little more room for the memory tiering algorithm to scale as more physical DRAM is available. With 1TB DRAM, this means we can run roughly 160 VDI sessions (leaving some room for overhead memory). By adding an NVMe device, which adds 1TB of memory to the system, we can run 160 more sessions, for a total of 320 VMs.

Figure 11. Memory Tiering allowed us to double the VM density with a less than 6% performance loss



The EUX score dropped from 8.7 to 7.9 (figure 12) when increasing from 160 VMs/1TB DRAM to 320 VMs/2TB DRAM. This is because the CPU was running at 1.5x turbo boost in the 1TB DRAM case. Because the CPU was utilization was high in the 320 VMs case, it was running at a frequency that was close to its normal level.

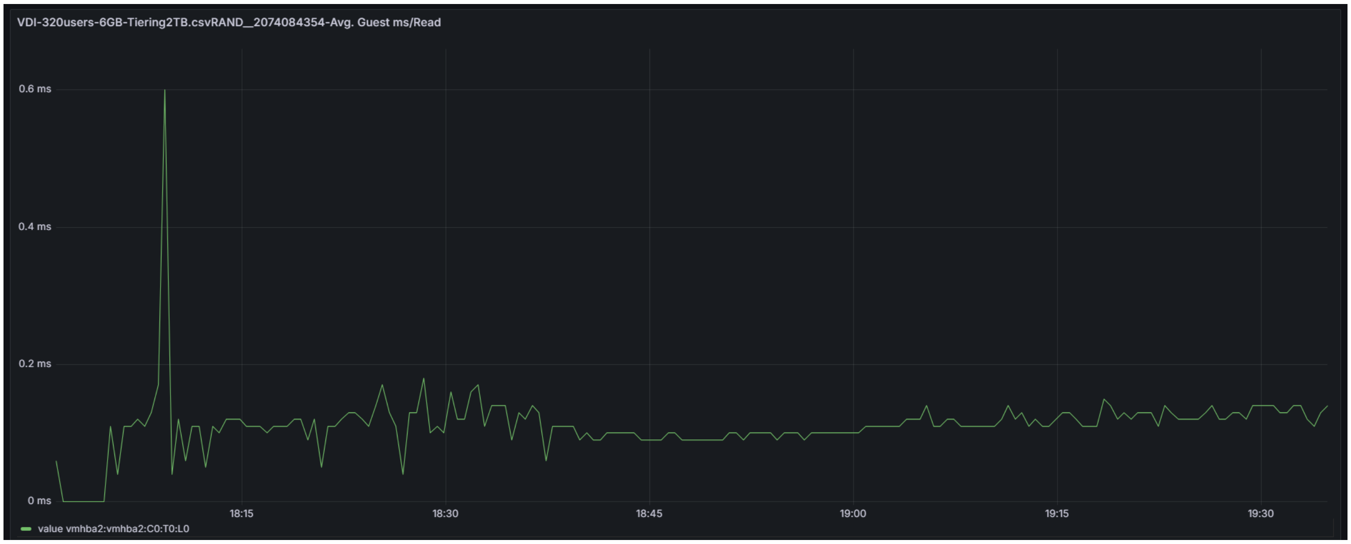
Figure 12. Comparable DRAM and Memory Tiering amounts (TB) showed only a 1.27% performance loss



With large pages enabled, there is no page sharing, but CPU utilization is lower. The EUX score was 8.3 with large pages enabled and the CPU utilization was 84.5%. When large pages were disabled, the EUX score shown in Figure 12 dropped to 7.9 because CPU utilization increased to 90%. That was enough to increase the ready time for the VMs and observe a consequent drop in the EUX score. This drop went from 8.3 in DRAM (2TB) with large pages to 7.8 in Memory Tiering with large pages disabled, resulting in a 6% performance loss. If we just consider small pages, the performance drop is from 7.9 to 7.8, only about a 1% drop.

This low performance drop correlates very well with a low-tier device latency of only 100 microseconds, because Memory Tiering read bandwidth is well below 100MBps, as shown in figure 13.

Figure 13. Read latency for Memory Tiering is well below 100 microseconds



Multi-node vSAN testing

For the multi-node tests, we used a 3-host, RAID 5, VMware® vSAN™ enterprise storage architecture (ESA) cluster. Each host in the cluster had 4 NVMe devices dedicated to vSAN with a 5th NVMe device used for memory tiering in the applicable test configurations.

> **Note:** The single-node configuration used processors with 56 cores per socket, whereas the vSAN multi-node VDI tests used vSAN systems with 32 cores per socket.

Table 3. Hardware and software used for multi-node vSAN testing

Component	Configuration
System under test (SUT)	Dell PowerEdge R660 with dual-socket Intel Xeon 6430 Processors (32 cores per socket) and 512GB DRAM
NVMe tier device	Dell-Ent-NVMe-CM6-MU-3.2TB-PCIe-3052360MB-SSD
SUT storage	VMware vSAN in a RAID 5, 3-host cluster with 4 SSDs: Dell-Ent-NVMe-CM6-MU-3.2TB-PCIe-3052360MB-SSD
VDI VMs	2 vCPUs, 6GB, Windows 11
Benchmark	Login Enterprise 5.14.7; 9 login sessions per minute across 3 hosts with a 30-minute test duration
Software for provisioning instant clones	Omnissa Horizon 8 2312

We configured the Login Enterprise workload using Windows 11 VMs and provisioned them using instant clones with Omnissa Horizon. As before, we tested the knowledge worker profile with a set of tests for each of the instant clone provisioning schemes.

ModeB performance test results

The first set of multi-node tests used the default mode (ModeB) of provisioning instant clone VMs in Omnissa Horizon, which did not result in a high level of page sharing across the VDI VMs.

Figure 14 shows the density of VDI VMs achieved, while figure 15 shows the performance results with 512GB DRAM, 1TB DRAM, and 1TB NVMe.

Figure 14. We ran double the number of VDI VMs using Memory Tiering with only an 8% performance loss

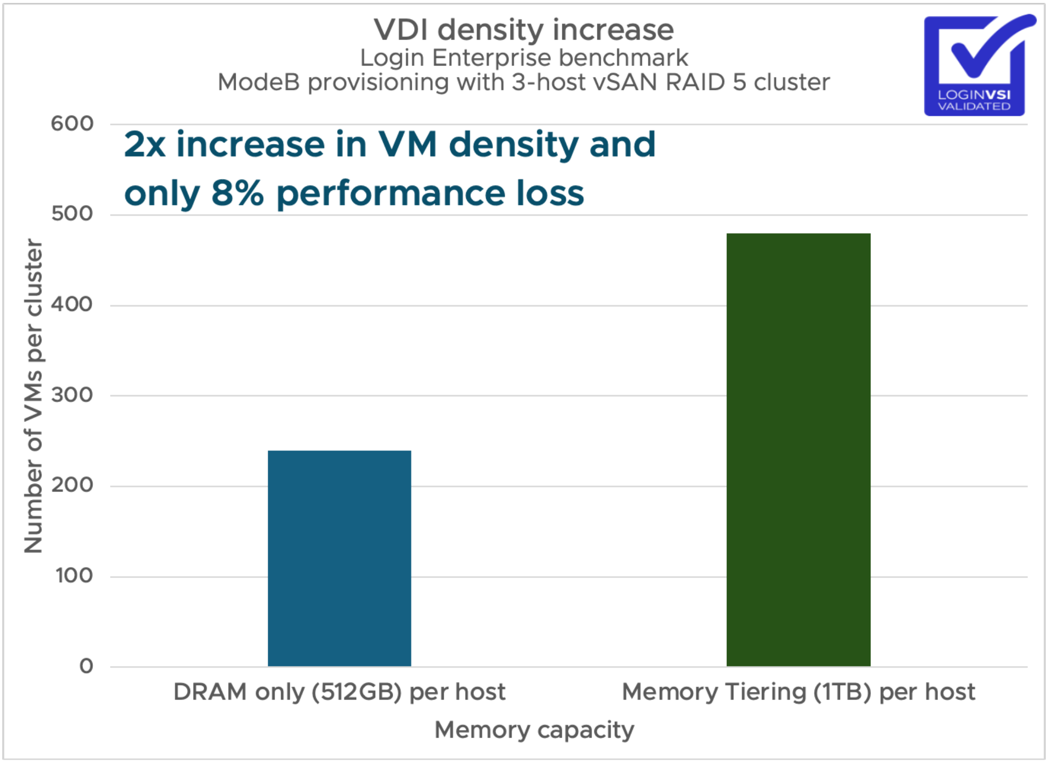
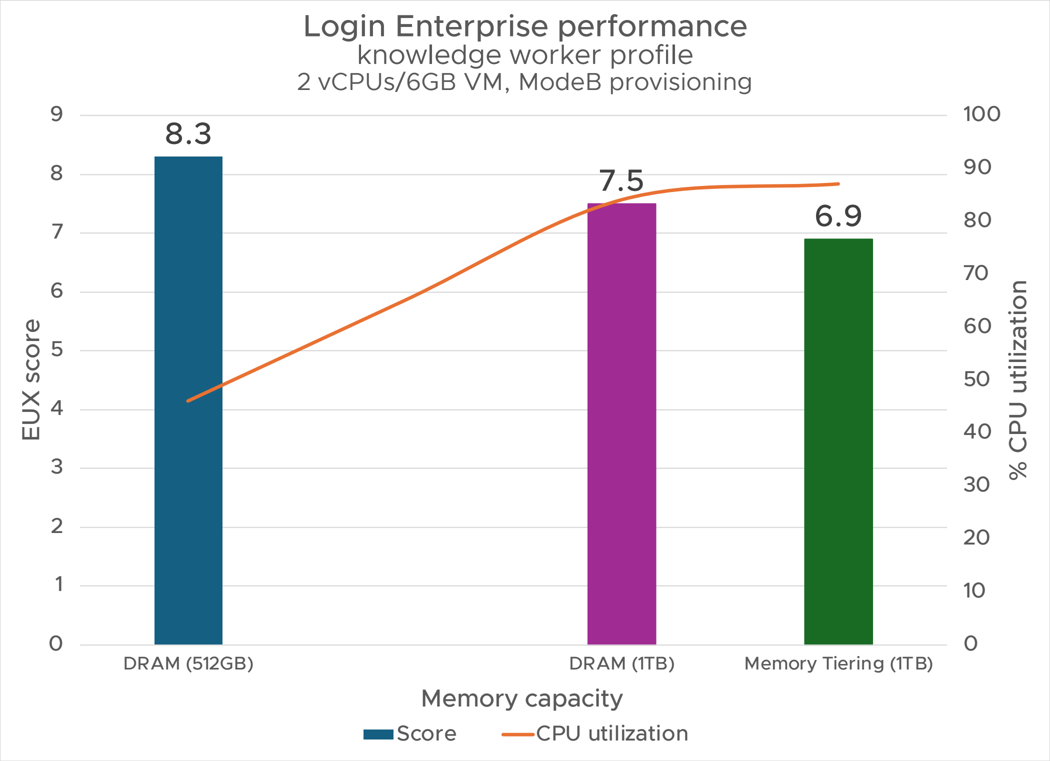


Figure 15. The EUX score decreased by only 8% with Memory Tiering



The tests show that Memory Tiering allows us to double the VM density by using an additional 512GB of NVMe in a 1:1 memory configuration, which results in just an 8% drop in EUX score performance relative to the 1TB DRAM case.

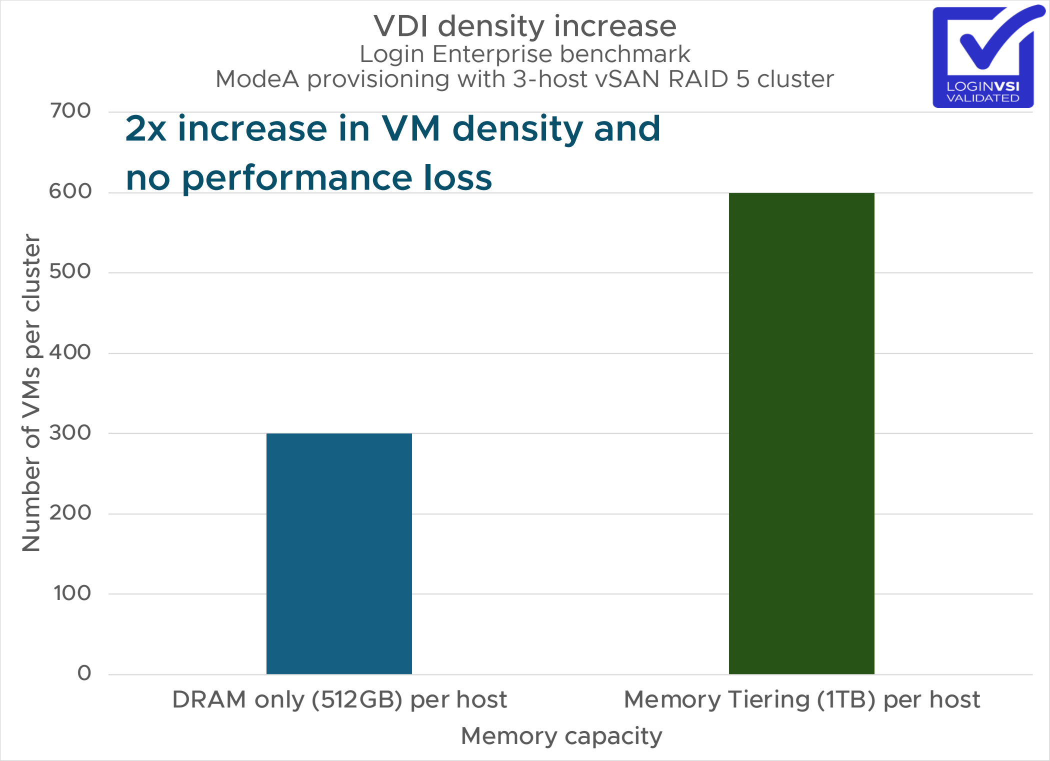
As in the single node case, when we disabled large pages for the DRAM (1TB) case, there was no difference in the EUX score; however, there was an increase in CPU utilization from 84% to 87%.

In the vSAN configuration, the NVMe device's read bandwidth and latency for ModeB tests exhibit similar behavior to that observed in the single node case, with read bandwidth values in the range of 200MBps and NVMe device latencies showing a similar trend. These latencies start around 100 microseconds, increase up to 400 microseconds as VDI sessions are added, and then drop back to 200 microseconds during the steady state portion of the test.

ModeA performance test results

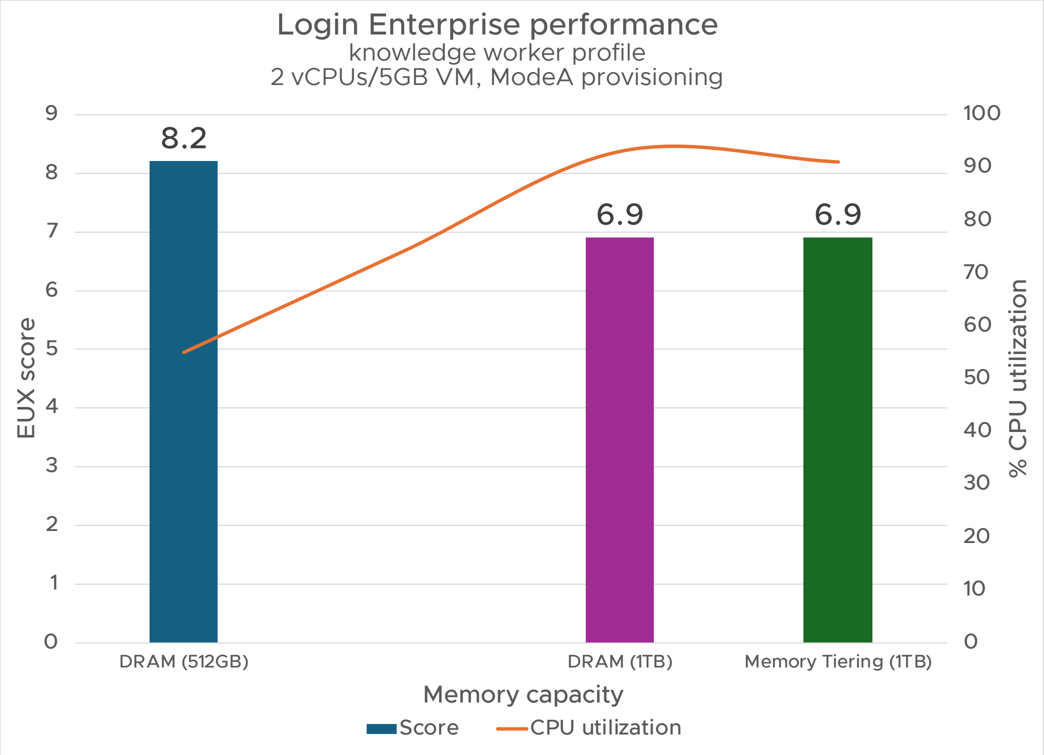
In a second set of tests, we increased the number of VDI VMs to 200 per host, or 600 for the entire vSAN ESA R5 cluster, and ran the Login Enterprise benchmark with 5GB instant clones and ModeA provisioning (see figure 16).

Figure 16. We doubled the number of VDI VMs per cluster when each host used 1TB Memory Tiering with no performance loss



EUX scores for the DRAM (1TB) test case and the Memory Tiering (1TB) were 6.9 and 7.0 respectively, with the CPU utilization in both cases over 90% (see figure 17). With this higher density of VMs, Memory Tiering device latencies were as high as 150 microseconds and the Memory Tiering device bandwidth hovered around or slightly above 200MBps. Finally, the active memory for this density of VMs was as high as 320GB, or 62.5% of the DRAM available to the Memory Tiering configuration.

Figure 17. Performance remained almost the same in the 1TB DRAM vs Memory Tiering cases



Enterprise platform software performance with VMmark

VMmark is a benchmark designed for evaluating the performance and scalability of virtualized environments, particularly virtualized datacenters.

We used VMmark 3.1 to deploy a diverse set of workloads across VMs running on a single server. VMmark bundles commonly virtualized applications (standby, DVD Store, and Weathervane) into predefined units called "tiles." A tile consists of 19 Linux VMs, running various workloads, configured with vCPU sizes ranging from 1 vCPU–8 vCPUs and memory sizes from 4GB–250GB.

We used the large memory configuration of VMmark. In this configuration, we changed the database VM size from 32GB to 250GB, increased the database size from 100GB to 300GB, and increased the think time from 1 second to 1.5 seconds to reduce CPU saturation and ensure the benchmark was more memory intensive than compute intensive.

Table 4. Hardware and software for the VMmark benchmark tests

Component	Configuration
System under test (SUT)	Dual-socket Intel Xeon 8592 Processors with 64 cores per socket and 1TB or 2TB DRAM
SUT NVMe tier device	Samsung PM9A3 3.84TB
SUT storage	SAN storage
Memory per tile	376GB
vCPUs per tile	31
Benchmark	VMmark 3.1

The VMmark score (shown in figure 18) is calculated by aggregating the application throughput metrics to create a single benchmark score after normalizing the data based on the weight of each application. The DVD Store (DS3) portion of the workload is weighted toward the overall VMmark score, while the Weathervane portion is weighted toward the quality-of-service (QoS) measurement. In our testing with VMmark, we loaded tiles on the system until QoS failures occurred.

Figure 19 shows the number of tiles and total number of VMs that VMmark supported in each configuration: 57 VMs within 3 tiles for the DRAM only (1TB) configuration, and 114 VMs within 6 tiles for the Memory Tiering (2TB) configuration. The latter configuration successfully ran 2x more VMs than the DRAM only (1TB) configuration.

Note: The active working set of VMmark for four tiles is about 500GB

Figure 18. We doubled the number of VMs as we went from 1TB of DRAM to 2TB Memory Tiering

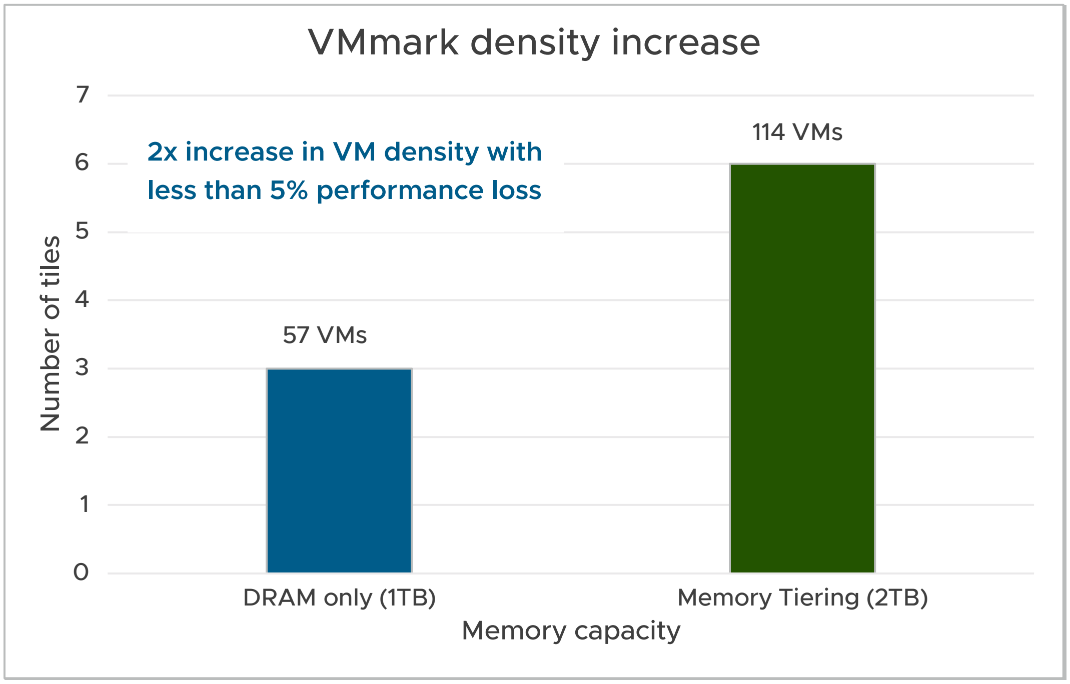
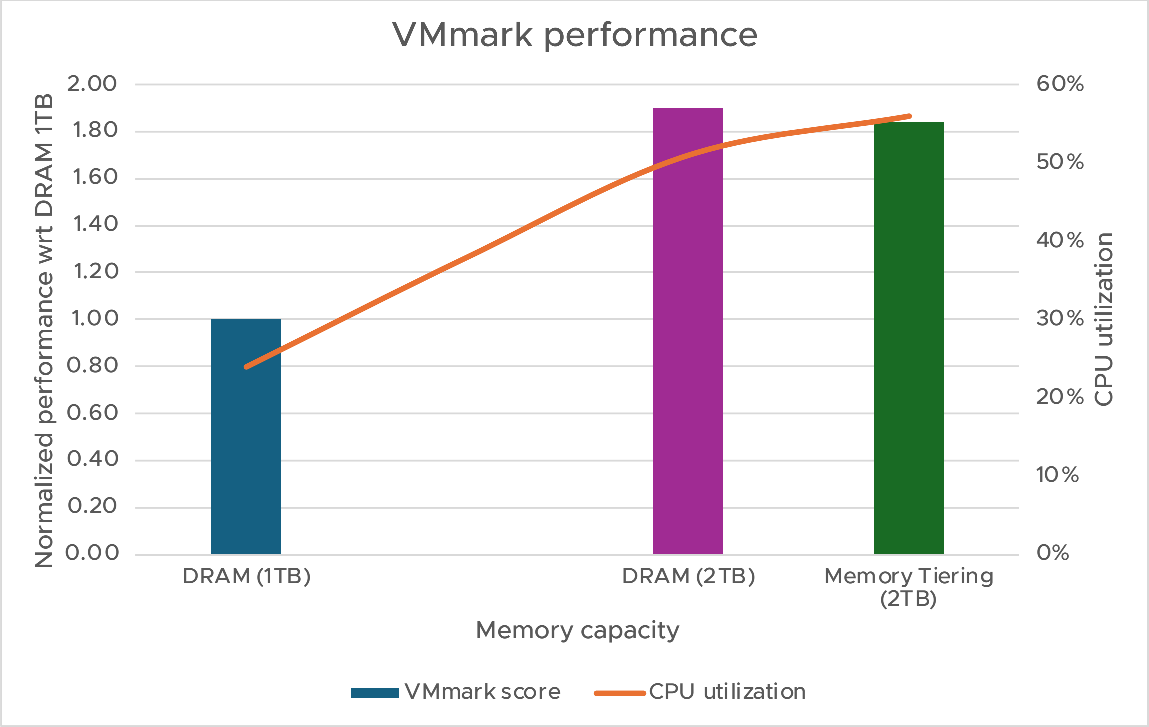


Figure 19 details the performance comparison of the three configurations tested: the baseline with DRAM (1TB), 2x baseline with DRAM (2TB), and Memory Tiering (2TB). This shows the overhead for Memory Tiering in terms of both performance and CPU cost. While CPU utilization was limited by the lack of DRAM resources in the baseline, the server was much higher utilized when there was more memory available. Performance with the Memory Tiering case is less than 5% lower than the DRAM (TB) case, even though the host with Memory Tiering has only 1TB of DRAM and is using an NVMe device for another 1TB of memory.

Figure 19. Memory Tiering allowed us to double the number of VMs with only a 5% performance loss



Figures 20 and 21 show the read throughput and read latency for the NVMe device on the VMmark host during the test. The read throughput is a bit higher than recommended, but because the latency remains very low at around 100 microseconds, performance remains good.

Figure 20. Read throughput of the NVMe device

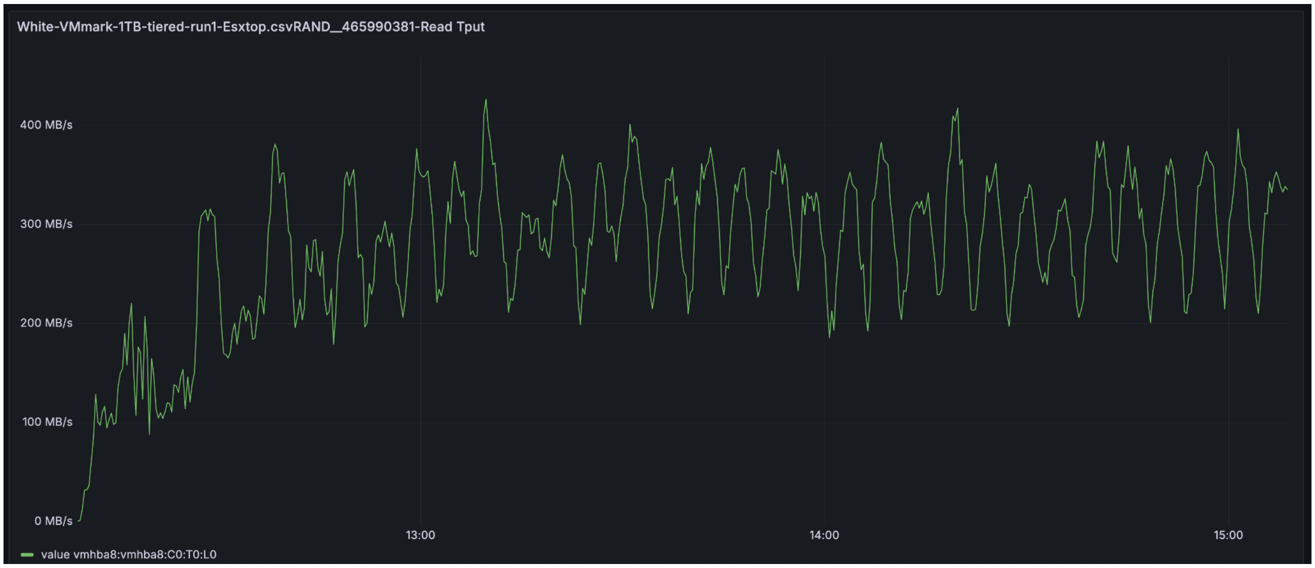
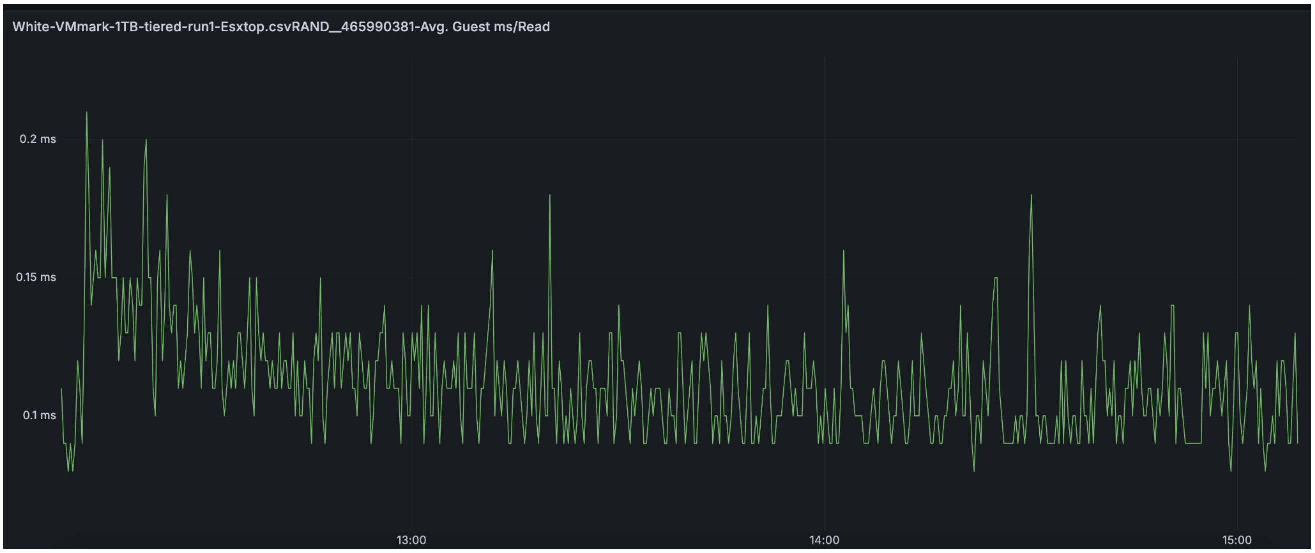


Figure 21. Read latency of the NVMe device



Database workload performance

Database workloads are performance-intensive, requiring CPU, disk, and memory resources. This characteristic makes them good for demonstrating the performance of Memory Tiering. We chose some databases typically seen in VCF deployments: Microsoft SQL Server 2022, Oracle Database 21c, and MySQL.

SQL Server 2022 with HammerDB

Microsoft SQL Server is a popular database used on VCF that supports a wide variety of applications. We tested its performance with Memory Tiering using HammerDB, a common database benchmark capable of simulating different workload types depending on the configuration used.

Table 5. Hardware and software used for the SQL Server/HammerDB tests and their configurations

Component	Configuration and version
System under test (SUT)	Dell PowerEdge R760 with a dual-socket Intel Xeon 8480 (56 cores per socket), 512GB and 1TB DRAM
SUT NVMe device	Dell_Ent_NVMe_P5620_MU_1.6TB
SUT storage	P5620 NVMe SSDs for database and database log disks
VM	8 vCPUs/80GB Windows Server 2022 VM running SQL Server 2022
Benchmark	HammerDB 5.0 (TPC-C profile)
Benchmark configuration	1000 warehouses with 125 virtual users and zero keying and think time; ramp_up = 15 mins; run_time=10 mins; total time = 25 minutes

Memory Tiering allowed us to double the VM density, as shown in figure 22. We were only able to run up to 6 VMs on an ESX host with 512GB of DRAM—the SQL Server transactions that were part of the HammerDB workload timed out if we tried to run more. When we boosted the memory capacity to 1TB using Memory Tiering, the number of VMs we could run on the host doubled.

Figure 22. VM density improved 2x with less than a 10% performance loss

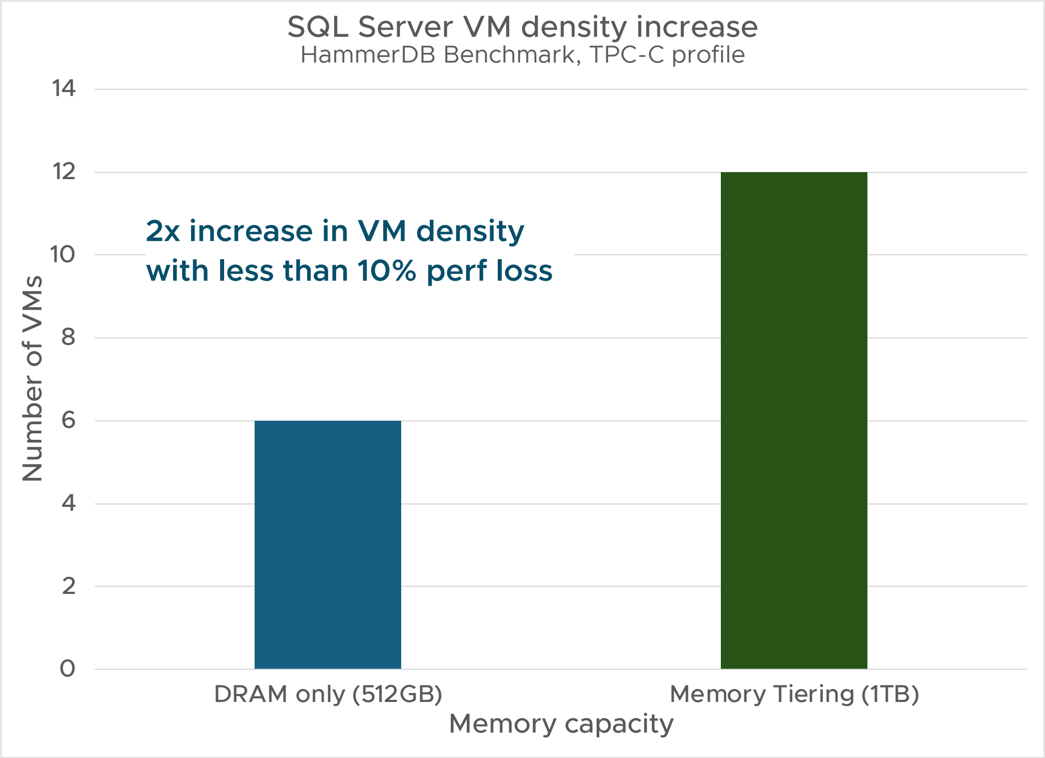


Figure 23 shows the performance increase from DRAM (512GB) to DRAM (1TB) was 1.6x. It was not a linear increase due to factors not related to Memory Tiering. The drop in performance from DRAM (1TB) and Memory Tiering (1TB) was less than 10%.

The active memory was about 256GB when the benchmark was running in a steady-state measurement period. The CPU cost dropped because the VMs were waiting for NVMe to service DRAM misses.

Figure 23. Microsoft SQL Server performance as measured with HammerDB

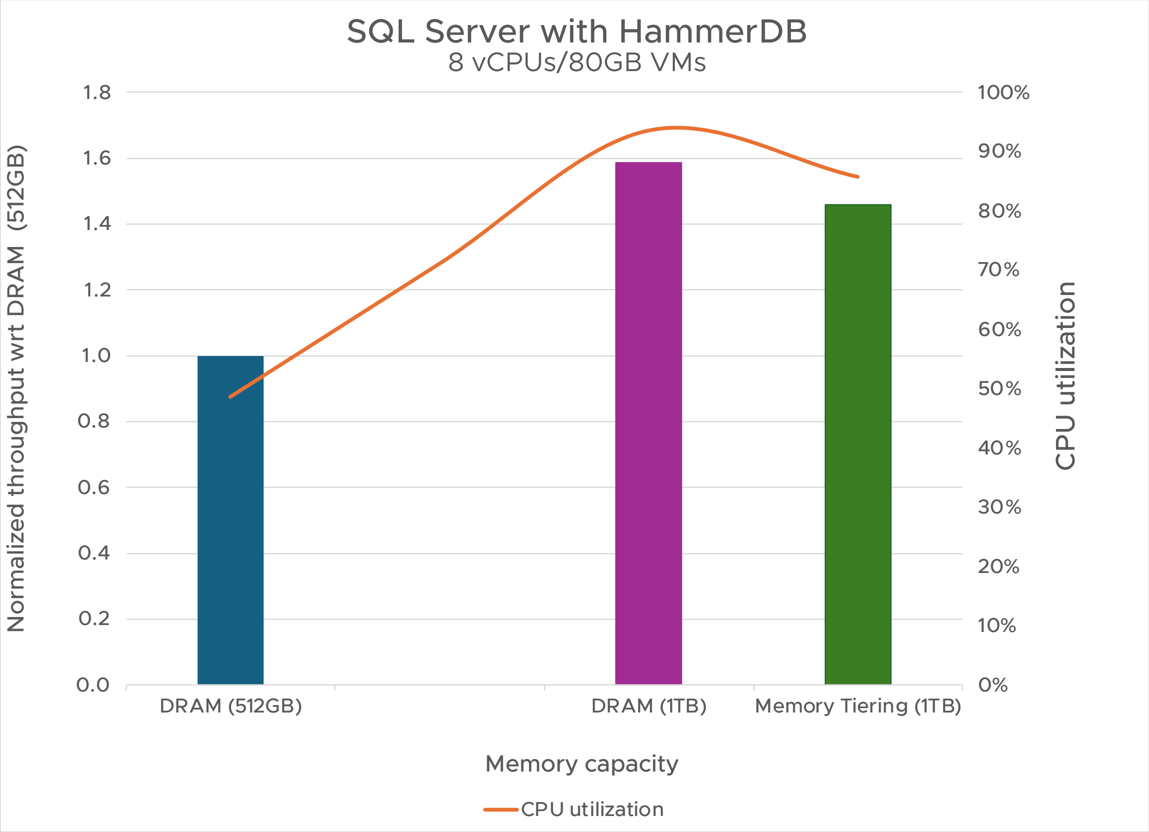
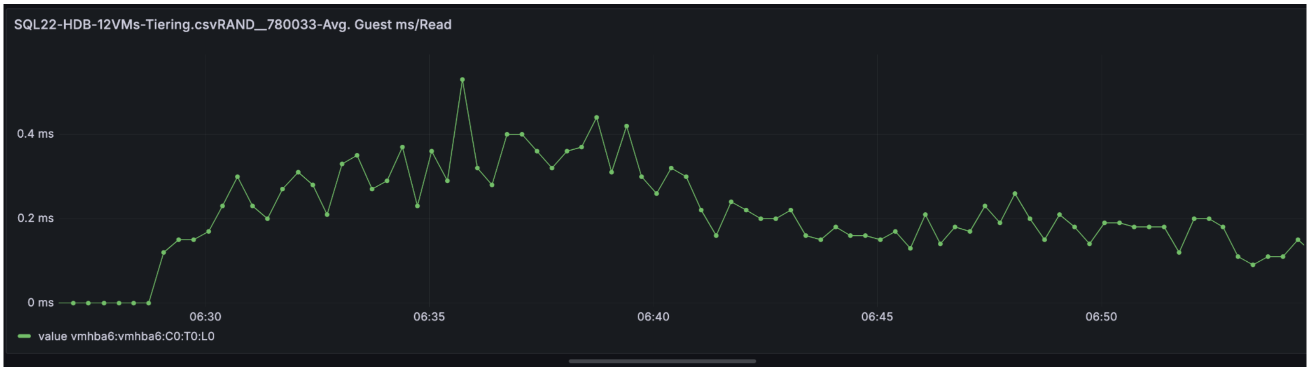


Figure 24 shows the NVMe device's read latency during the full course of the HammerDB test, which starts out with a ramp-up phase. During the initial ramp-up, the NVMe device latency hovered around 300–400 microseconds and then stabilized at approximately 200 microseconds during the run phase of the benchmark.

Figure 24. Read latency of the NVMe device


SQL22-HDB-12VMs-Tiering.csvRAND__780033-Avg. Guest ms/Read
0.4 ms
0.2 ms
0 ms
06:30
06:35
06:40
06:45
06:50
value vmhba6:vmhba6:C0:T0:L0

Figures 25 and 26 show the active memory and throughput for the VMs during the HammerDB test run. You can see that when active memory was more than 50% of DRAM during the ramp-up phase, the SQL Server transactions per minute (TPM) for various VMs dropped. After the ramp-up phase ended and the run phase started, active memory settled in at about 50% of DRAM (about 256GB) and the TPM rose and stabilized.

Figure 25. Active memory for the VMs during the test run, as shown in the VCF Operations dashboard

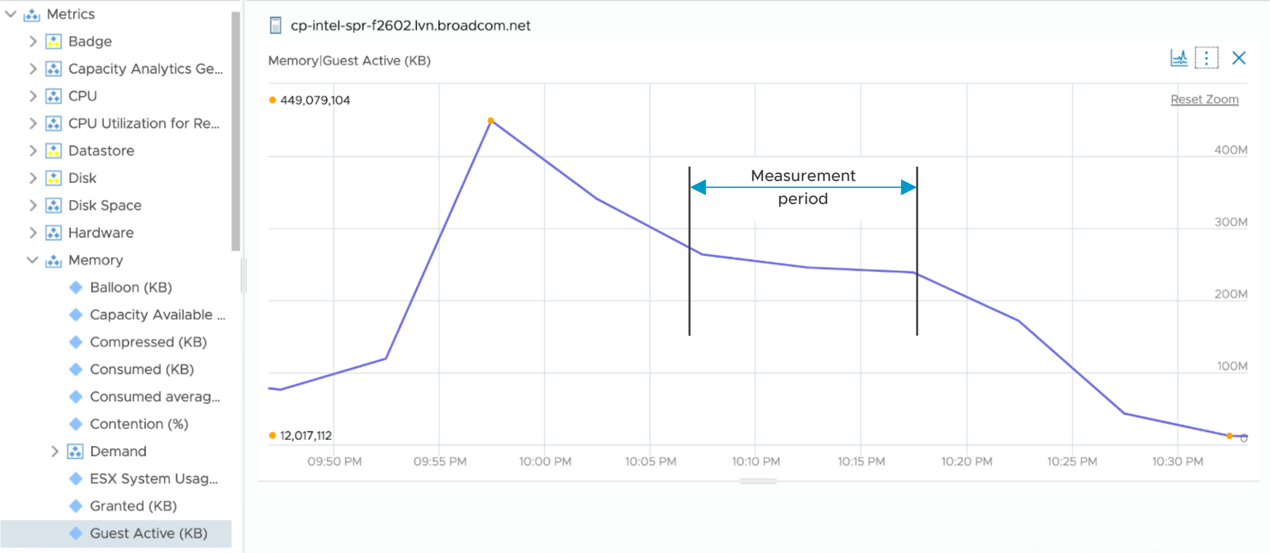
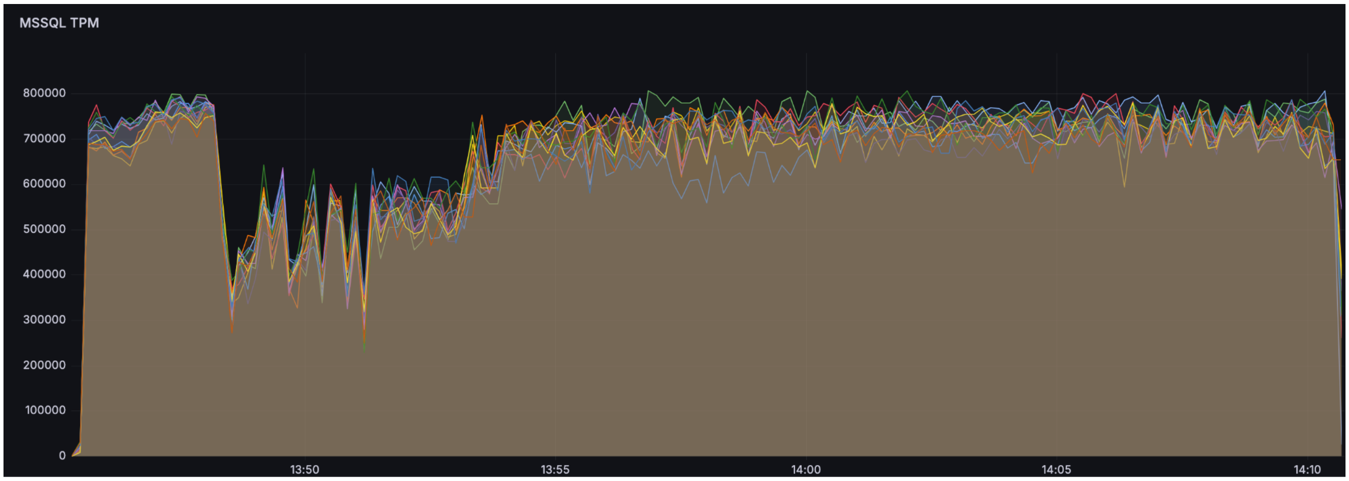


Figure 26. Throughput (transactions per minute) for the VMs during the test run



Oracle Database with DVD Store

We tested with Oracle Database 21c on Oracle Enterprise Linux 8.8 using the DVD Store 3.5 workload. We used 600 users per VM, and a think-time set to 5 seconds on a database of about 200GB. This high number of users and realistic think time placed high stress on both memory and CPU resources.

Table 6. Hardware and software configurations

Component	Configuration and version
System under test (SUT)	Single socket AMD EPYC 9755 (128 cores) with 768GB DRAM
SUT NVMe device	Dell_Ent_NVMe_CM6-MU-3.2TB-PCIe-3052360MB-SSD
SUT storage	3x local NVMe: Dell-Ent-NVMe-P5500-3.84TB
VM	16 vCPUs/192GB Oracle Enterprise Linux 8.8 VM running Oracle Database 21c
Benchmark	DVD Store 3.5
Benchmark configuration	600 users per VM with 5 seconds of think time

We tested three scenarios:

- Baseline case with 768GB DRAM
- Baseline case with 1.5TB DRAM
- Memory Tiering case with 768GB DRAM and 768GB NVMe for a total of 1.5TB memory

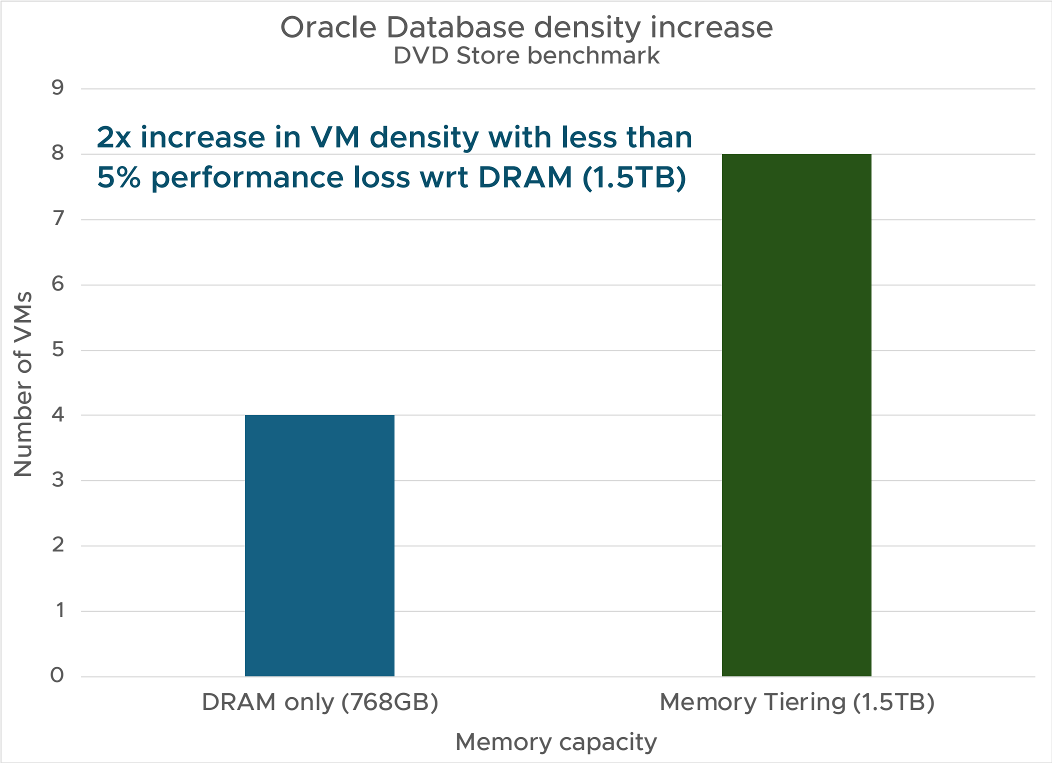
These configurations allowed us to see how close Memory Tiering compared in performance against the DRAM-only baseline cases.

We assigned 192GB of RAM to each VM, and then added enough VMs to each scenario to fully commit the memory:

- For the DRAM only (768GB) scenario, we used 4 VMs (192GB x 4 = 768GB)
- For the DRAM (1.5TB) and Memory Tiering (1.5TB) scenarios, we used 8 VMs (192GB x 8 = 1.5TB)

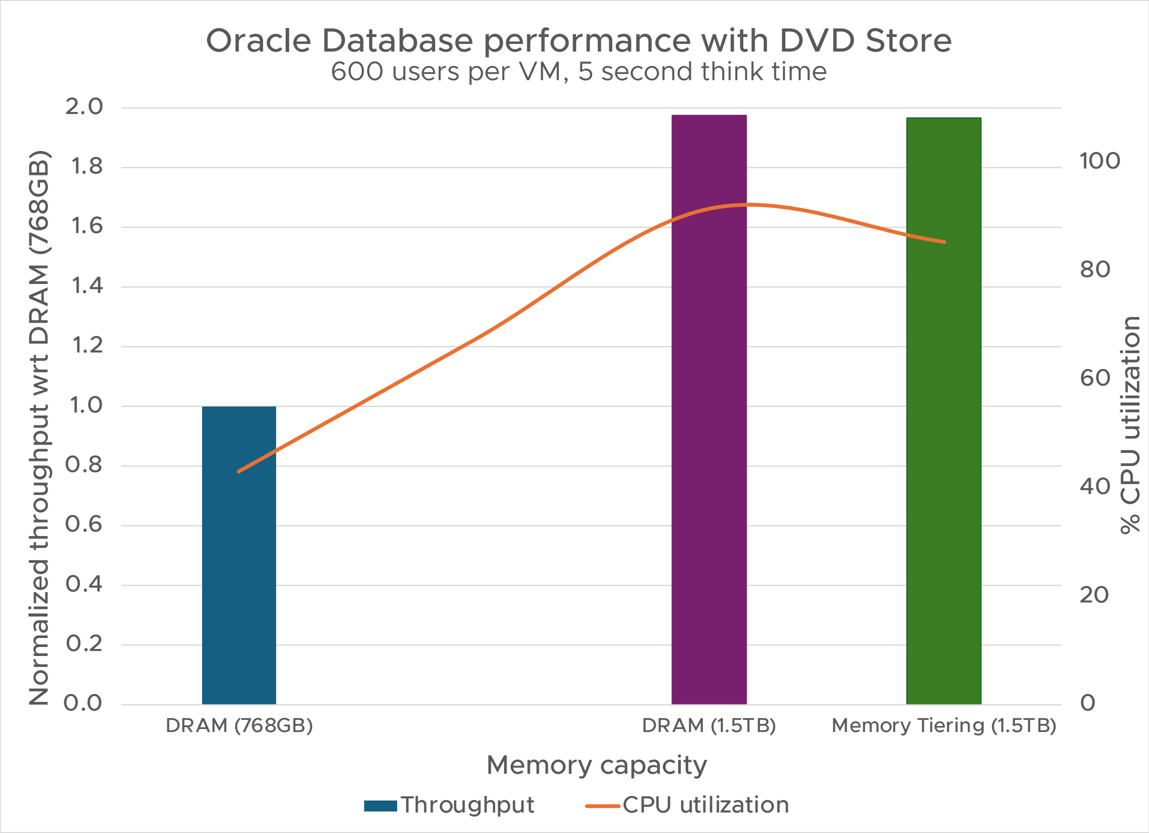
Results showed (figure 27) that we could increase the number of VMs on a host from 4 to 8 with less than a 5% loss in performance compared to the 1.5TB DRAM vs 1.5TB Memory Tiering configuration (figure 28).

Figure 27. VM density increase with Memory Tiering



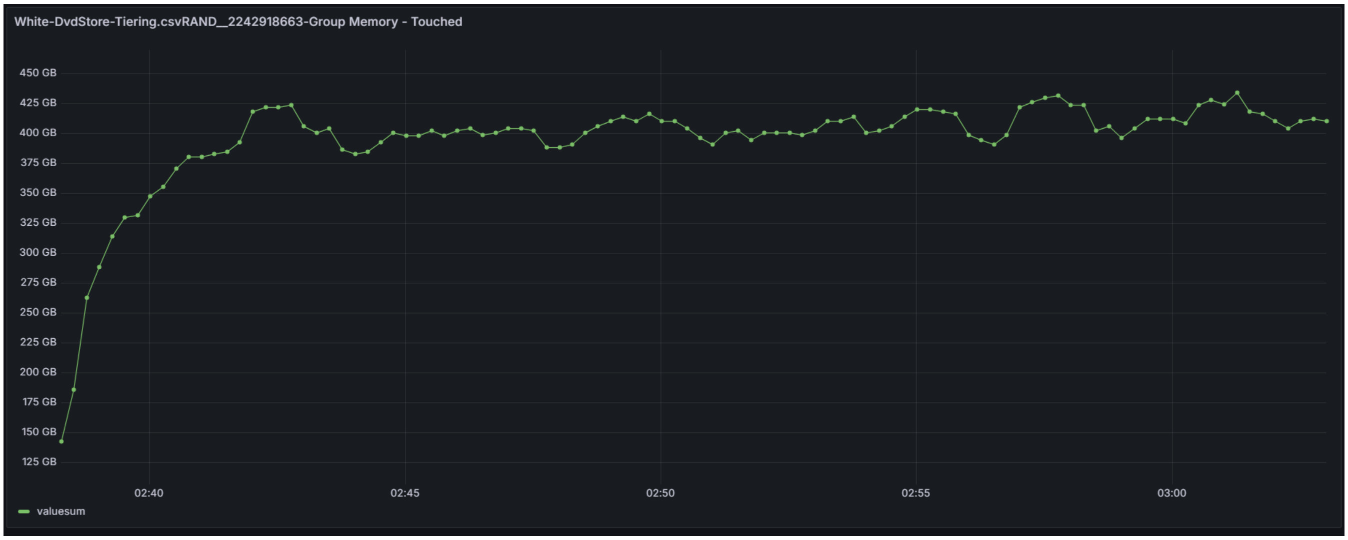
The core CPU utilization was limited to 43% in the 768GB DRAM baseline test case (figure 28), but using Memory Tiering allowed for the CPU utilization to reach 85%. This shows the ability of Memory Tiering to unlock the additional capacity of the host in cases where the host is memory constrained and there are available CPU cycles not being used.

Figure 28. Throughput vs CPU utilization for each memory capacity scenario



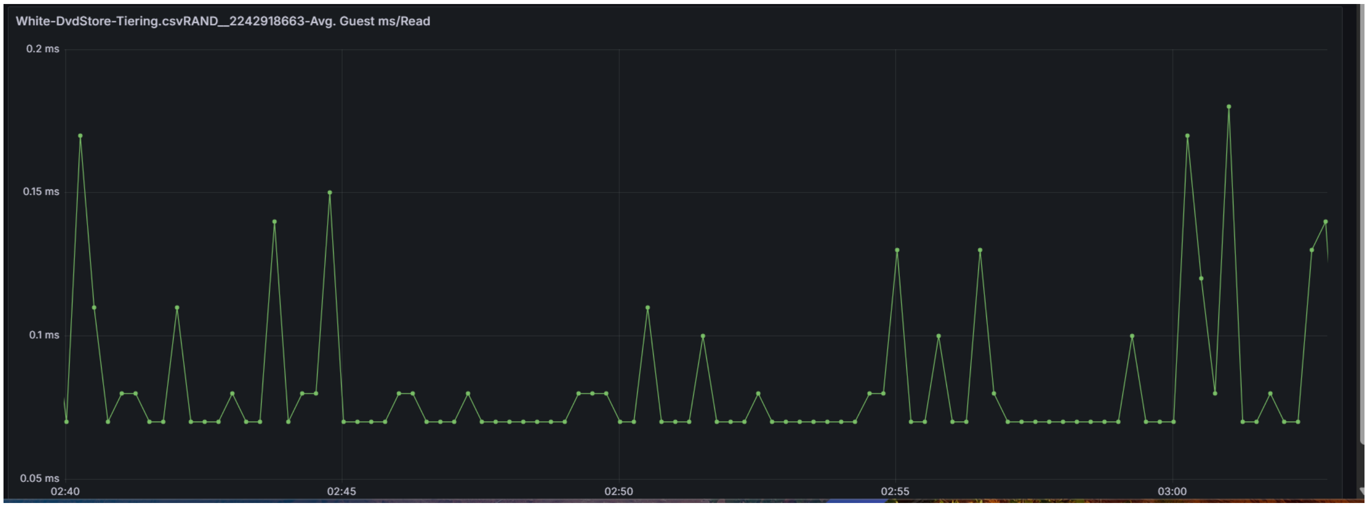
We observed the active memory, as measured by Touched Memory from esxtop, with the workload running across all 8 VMs, to average about 400GB (figure 29). This put it at just over 50% of the DRAM for the Memory Tiering test case, which is in line with the best practice of keeping active memory at 50% of DRAM.

Figure 29. Touched memory in esxtop averaged about 400GB



The NVMe device showed an average read latency of 86 microseconds during the Memory Tiering 1.5TB test run (figure 30). We observed the read bandwidth at around 35MBps, which was the cause of the low read latency and hence good performance.

Figure 30. Average read latency of the NVMe device



MySQL Database with HammerDB

Our last test measured MySQL database performance with the HammerDB benchmark. The hardware and software configurations are shown in table 7.

Table 7. Hardware and software configuration for the MySQL Database testing with HammerDB

Component	Configuration and version
System under test (SUT)	Dell PowerEdge R760 with a dual-socket Intel Xeon 8480 (56 cores per socket), 512GB and 1TB DRAM
SUT NVMe device	Dell_Ent_NVMe_P5620_MU_1.6TB
SUT storage	P5620 NVMe SSDs for database and database log disks
VM	14 vCPUs/60GB RHEL 9.4 VM running MySQL 8.0
Benchmark	HammerDB 5.0 (TPC-C profile)
Benchmark configuration	1000 warehouses with 125 virtual users and zero keying and think time; ramp_up = 15 mins; run_time=10 mins; total time = 25 minutes

As in the other database tests, we could double the VM density with less than a 5% performance loss (figures 31 and 32).

Figure 31. VM density increased 2x with the use of Memory Tiering

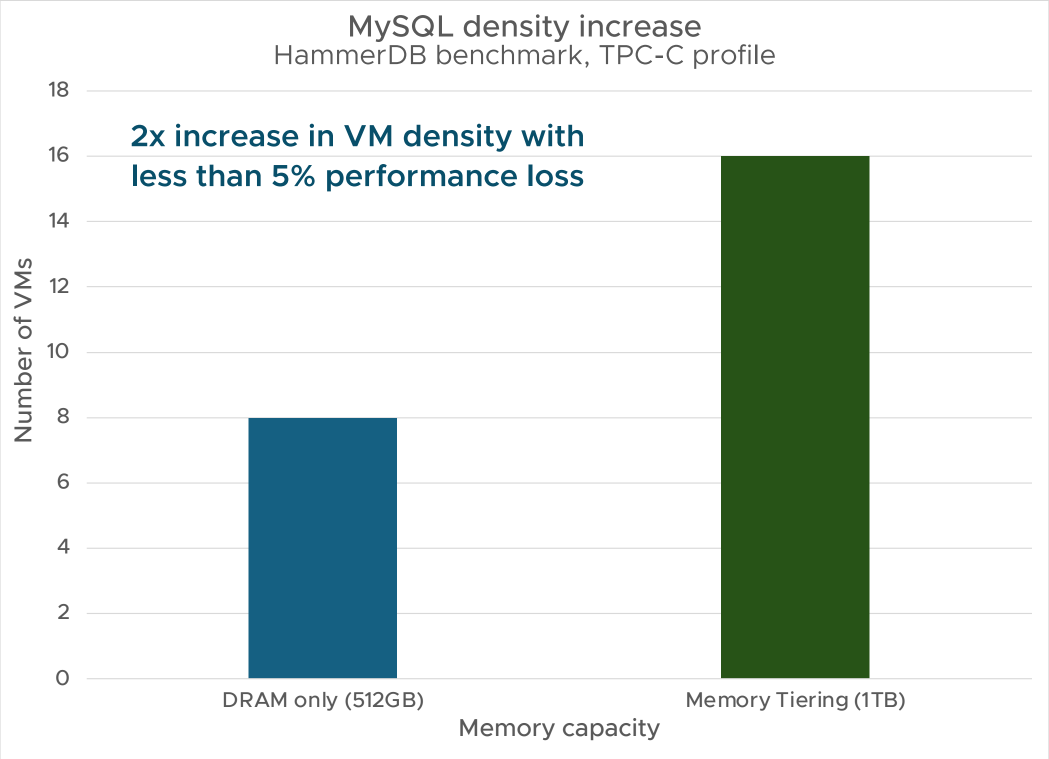
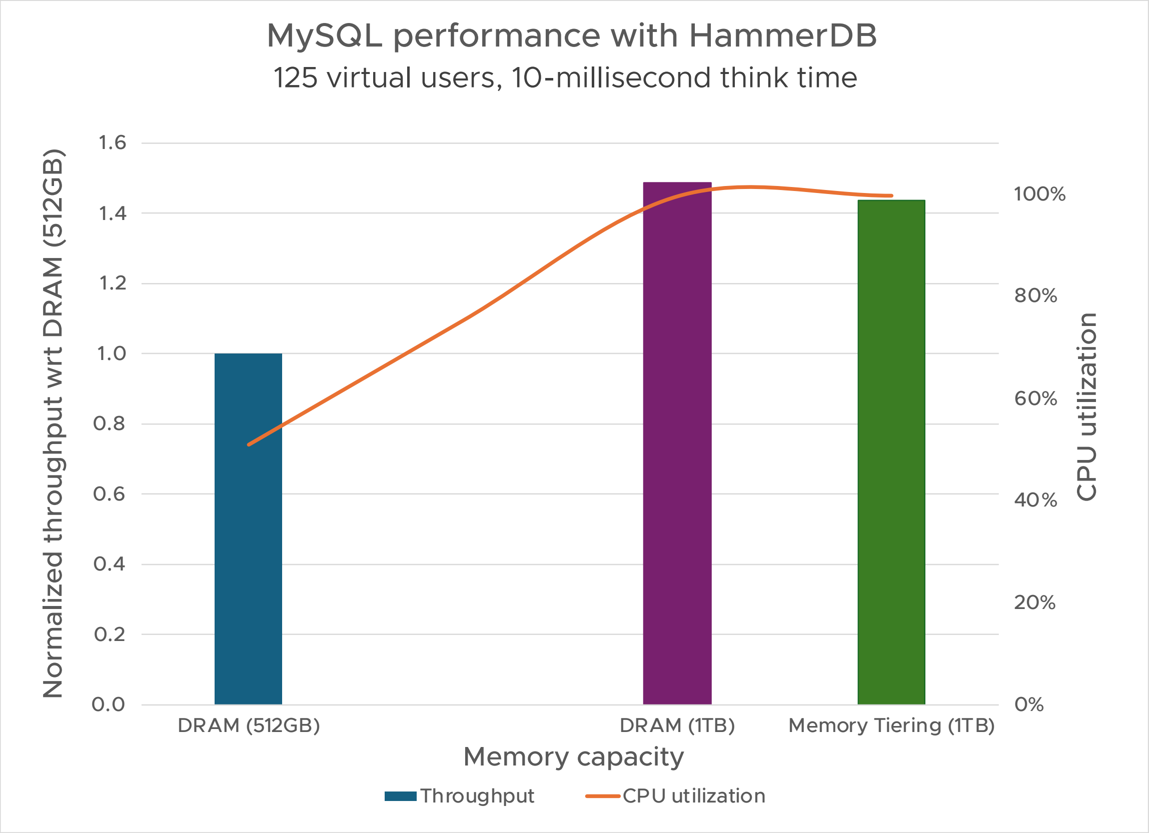
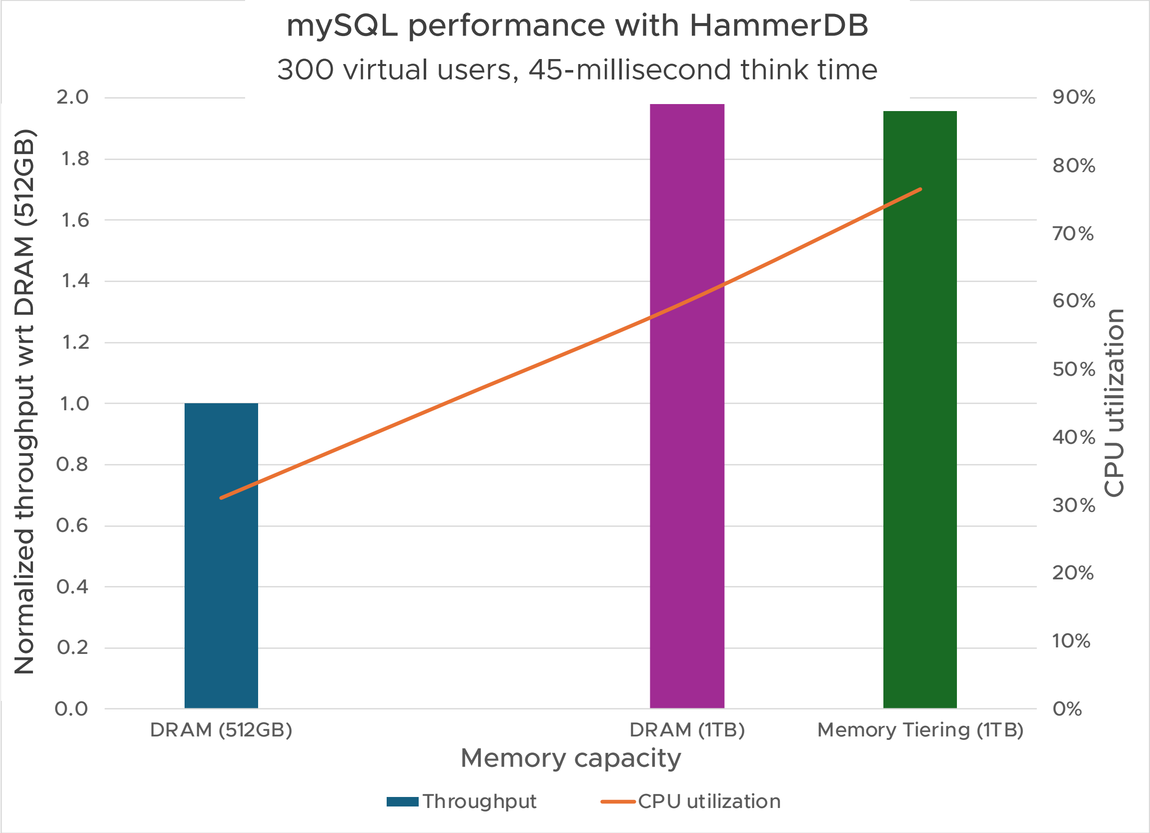


Figure 32. Configuration 1: Normalized throughput vs CPU utilization



In these MySQL experiments, we played with the “keyingandthinktime” settings of HammerDB to control CPU and active memory. Figure 33 shows that in the baseline case of DRAM (512GB), CPU utilization was high at 70%, because of the lower think time of 10 milliseconds. Active memory was around 55% of DRAM capacity 286GB. Doubling the number of VMs at DRAM (1TB) and Memory Tiering (1TB), we could run 224 vCPUs (14 vCPUs per VM x 16 VMs) and CPU reached near saturation, which explains the non-linear scaling. Even at this high CPU utilization, performance loss with memory tiering was less than 5%.

Figure 33. Configuration 2: Normalized throughput vs CPU utilization



We increased the think time to 45 milliseconds and increased the number of virtual users in the next experiment. This increased the active memory from 286GB to about 410GB (approximately 80% of DRAM capacity) and reduced the CPU utilization. For DRAM (512GB), CPU utilization was 45%. Since there was more CPU headroom, the 2x increase in VM density increased throughput by 2x as well. The turbo frequency was about the same across all configurations because the VMs were not fully saturated, unlike in the previous test. This also explains the linear scaling. An interesting point to note here is that even though active memory was 80% of DRAM capacity, the performance loss was very small. This was probably because the think time was high enough to absorb the latencies from NVMe.

vMotion performance implications in a Memory Tiering environment

VM performance remains unaffected during vMotion with Memory tiering. The migration may take longer because the pre-copy phase involves reading data from slower tiers, which slows down the transfer. Refer to this paper to learn more about vMotion internal mechanism.

Table 8. Hardware and software for vMotion performance testing

Component	Configuration and version
System under test (SUT)	Dell PowerEdge R750 with a dual-socket Intel Xeon 8380 (40 cores per socket), 1TB DRAM
SUT NVMe device	Dell_Ent_NVMe_P5600_MU_1.46TB
SUT storage	Dell EMC Unity 600 all-flash array
NICs	Mellanox 100GbE network adapter
VMs	4 VMs, each configured as 12 vCPUs/48GB RHEL 8.1 VM running Oracle 21c with 43GB SGA
Benchmark	HammerDB 5.0
Benchmark configuration	1000 warehouses with 60 virtual users and zero keying and think time

Note: We reduced the host memory to 164GB to conduct the experiment so that Memory Tiering was activated.

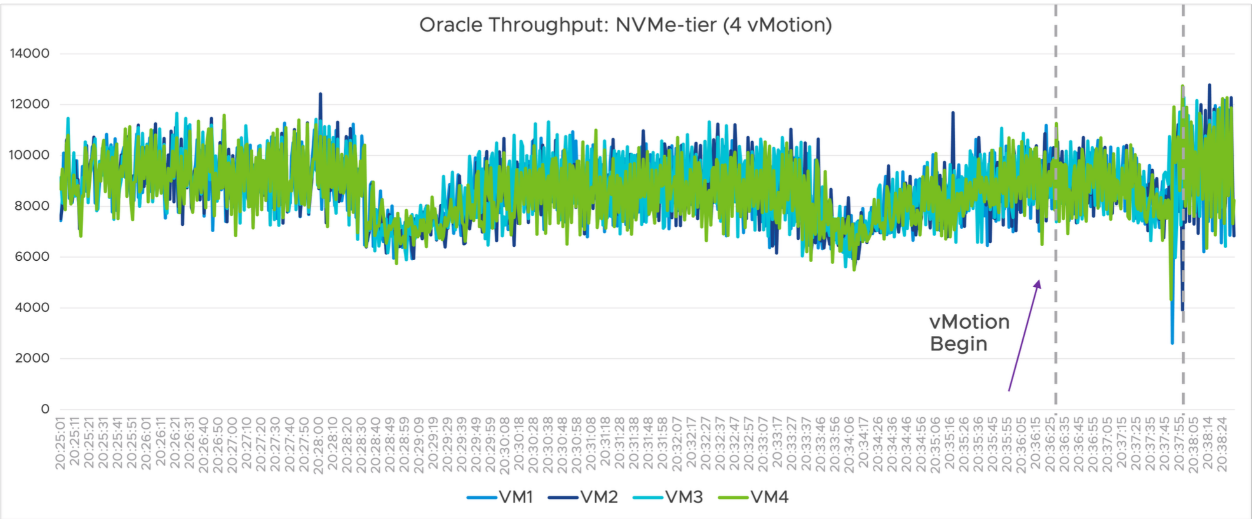
We looked at migration time, switch-over time, and guest penalty to measure the performance of vMotion in our tests:

- **Migration time:** The total time taken for the migration to complete.
- **Switch-over time:** The time during which the VM is quiesced (paused) during switchover from the source host to the destination host.
- **Guest penalty:** The performance impact (latency and throughput) on the applications running inside the virtual machine during and after the migration.

We modeled a host evacuation scenario where we migrated four typical-sized VMs at the same time.

Oracle throughput remained minimally impacted across all four VMs during the migration in the Memory Tiering configuration. We observed only one performance dip during switch-over phase, but VM downtime remained under 1 second.

Figure 34. Oracle throughput during 4 vMotion migrations with Memory Tiering enabled



The following table summarizes vMotion performance key metrics in both baseline and NVMe-tier scenarios.

Table 9. Key vMotion performance metric in baseline and Memory Tiering host evacuation scenarios

Scenario	Average migration time	VM downtime	Guest penalty
4-VM baseline (DRAM only)	23.5 seconds	< 1 second	< 5%
4-VM Memory Tiering	82 seconds	< 1 second	< 5%

The table shows that Memory Tiering increased the vMotion duration. However, VM performance was unaffected, because the slowdown occurred during the precopy phase, where cold memory pages are read from the slower NVMe device.

Using and monitoring Memory Tiering

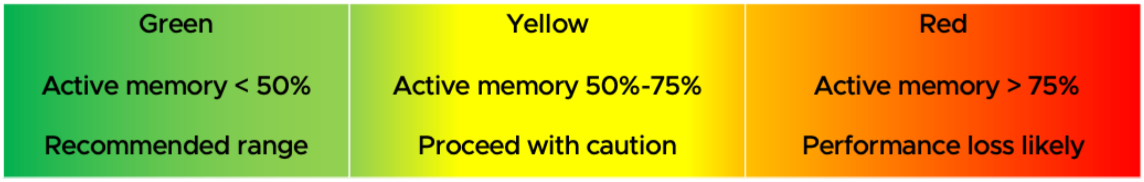
There are three key things you can do to ensure that Memory Tiering provides good performance for your running workloads:

- Monitor active memory
- Monitor the NVMe device's read latency
- Select a high quality NVMe device

Monitor active memory

Keeping active memory at 50% or lower of the **DRAM capacity** in the host is recommended. For some workloads we have found that it can be higher with no issues. In between 50% and 75% it is necessary to do some testing and monitoring of workload performance. Beyond 75% active memory we would expect to see significant performance loss in most cases. The table below summaries this with a green, yellow, red color coding for simplicity.

Figure 35. Monitor active memory to keep it in the green range

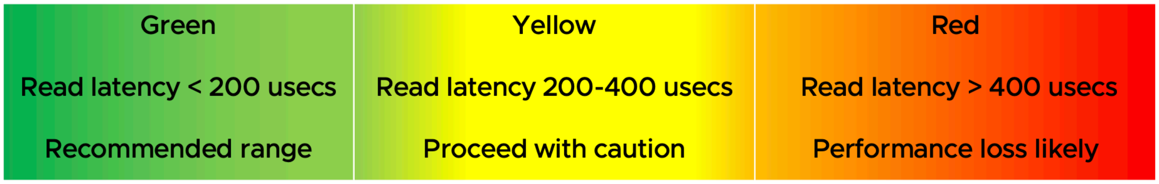


Green	Yellow	Red
Active memory < 50%	Active memory 50%-75%	Active memory > 75%
Recommended range	Proceed with caution	Performance loss likely

Monitor the NVMe device's read latency

The performance of the NVMe device is also important to monitor. If the device's read latency remains below 200usecs, then we expect performance of memory tiering to remain good. The potential for some issues exists when the latency grows and is between 200-400 microseconds. Once the NVMe tier device latency is above 400 microseconds, we expect to see performance impact on workloads. The color-coded chart below summarizes this guidance.

Figure 36. Monitor the NVMe device to make sure its read latency is in the green range



Green	Yellow	Red
Read latency < 200 usecs	Read latency 200-400 usecs	Read latency > 400 usecs
Recommended range	Proceed with caution	Performance loss likely

Select a high quality NVMe device

It is important to select an NVMe that has high endurance class D and high-performance class with more than 100,000 writes per second with (DWPD = 3) and larger capacity.

Use vCenter to find metrics

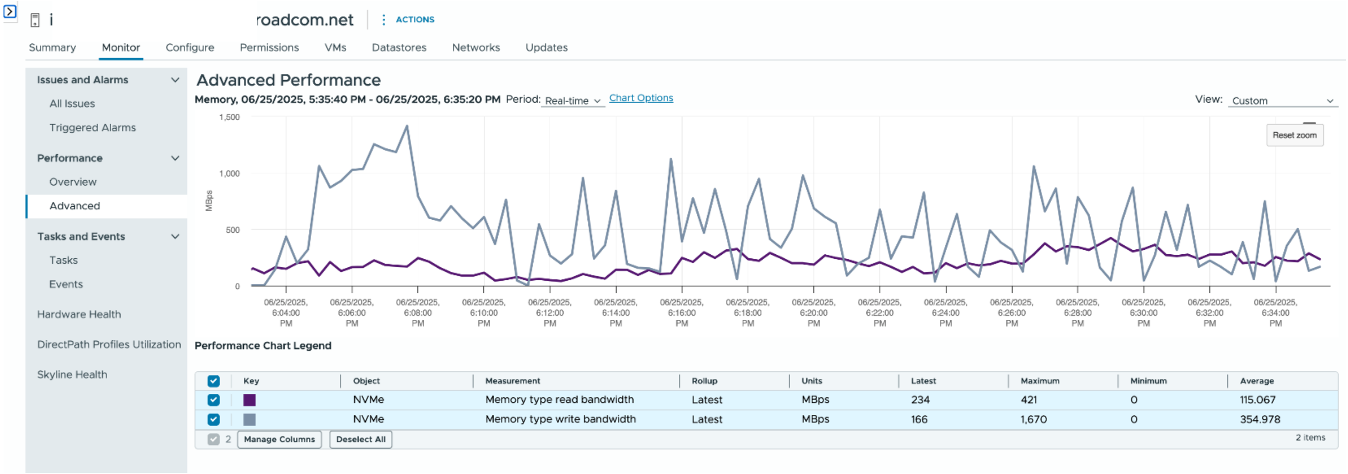
Below are some screenshots from vCenter showing where you can find various metrics.

Read and write bandwidth for host

Figure 37 shows a screenshot of the memory view in vCenter, found by selecting the memory read bandwidth and memory write bandwidth from **Chart Options** on the **Advanced Performance** page. The write bandwidth shows the cold pages being moved into NVMe and read bandwidth shows that if a page becomes active and is not present in DRAM, it is being fetched from NVMe. If NVMe read bandwidth is higher than 200MBps, you might start looking at device latencies.

> **Note:** Depending on what class of device you are using, latencies might be higher or lower than those shown here.

Figure 37. Advanced Performance memory view in vCenter with read and write **bandwidths** shown



Performance Chart Legend

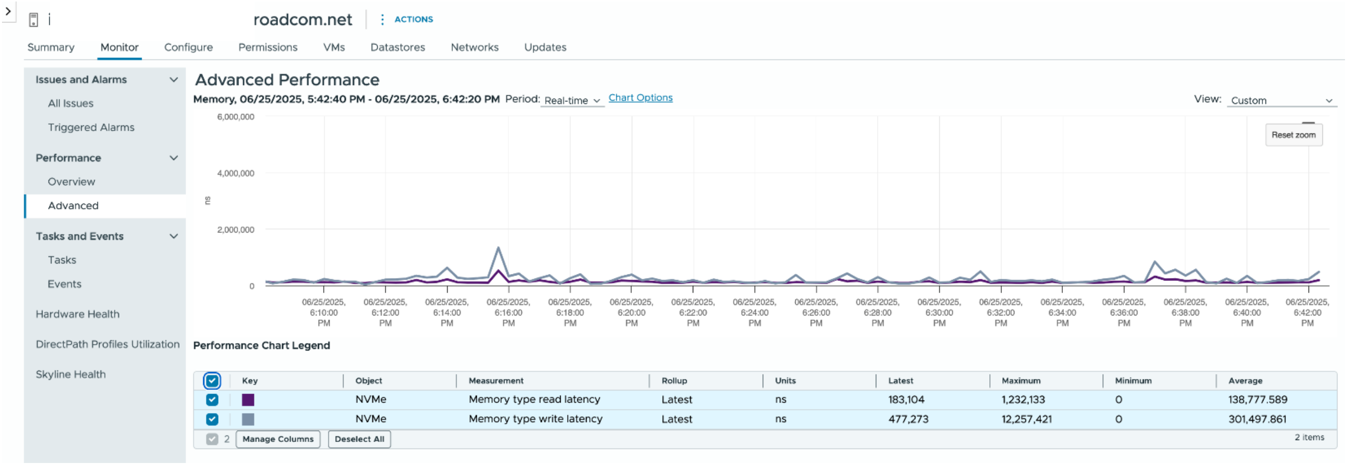
☑	Key	Object	Measurement	Rollup	Units	Latest	Maximum	Minimum	Average
☑	■	NVMe	Memory type read bandwidth	Latest	MBps	234	421	0	115.067
☑	■	NVMe	Memory type write bandwidth	Latest	MBps	166	1,670	0	354.978

☑ 2 Manage Columns Deselect All 2 items

NVMe device latencies for host

Figure 38 shows a screenshot of the memory view in vCenter, found by selecting the memory read latency and memory write latency from **Chart Options** on the **Advanced Performance** page. Read latencies stay well below 200 microseconds even at 400MBps read bandwidth because the NVMe device is a high-performance one. In contrast, some other NVMe drives show around a 300-microsecond read latency at a similar bandwidth. Monitoring latency is critical, because it directly affects workload performance.

Figure 38. Advanced Performance memory view in vCenter with read and write **latencies** shown



Performance Chart Legend

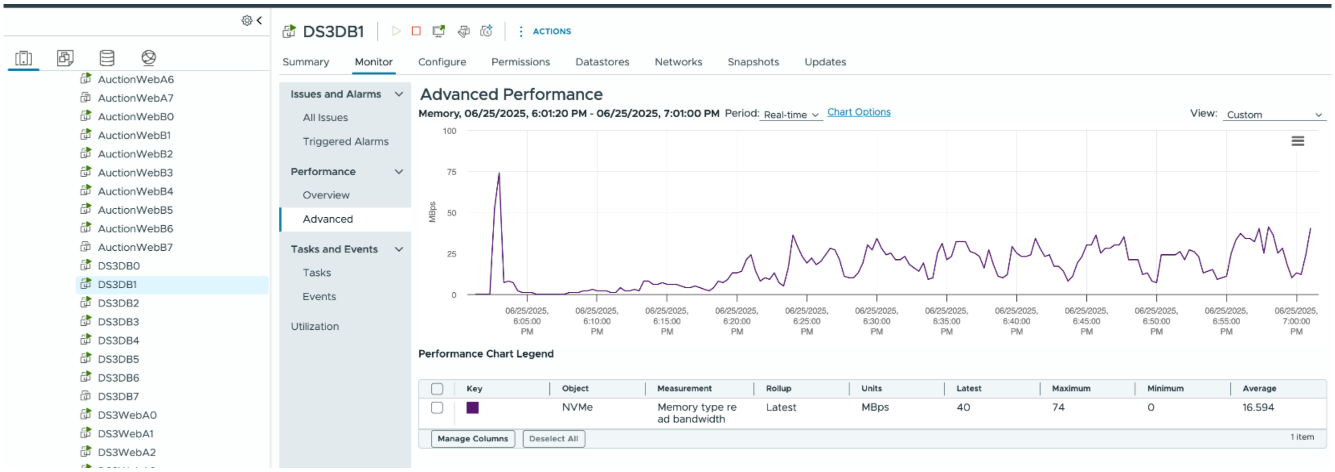
☑	Key	Object	Measurement	Rollup	Units	Latest	Maximum	Minimum	Average
☑	■	NVMe	Memory type read latency	Latest	ns	183,104	1,232,133	0	138,777.589
☑	■	NVMe	Memory type write latency	Latest	ns	477,273	12,257,421	0	301,497.861

☑ 2 Manage Columns Deselect All 2 items

Read bandwidth per VM

If you need to determine why the performance of a particular VM is low, look at the per-VM read bandwidth, as shown in figure 39. For multiple VMs, see figure 40.

Figure 39. Advanced Performance memory view in vCenter with read bandwidth shown for the selected VM



Advanced Performance

Memory, 06/25/2025, 6:01:20 PM - 06/25/2025, 7:01:00 PM Period: Real-time Chart Options View: Custom

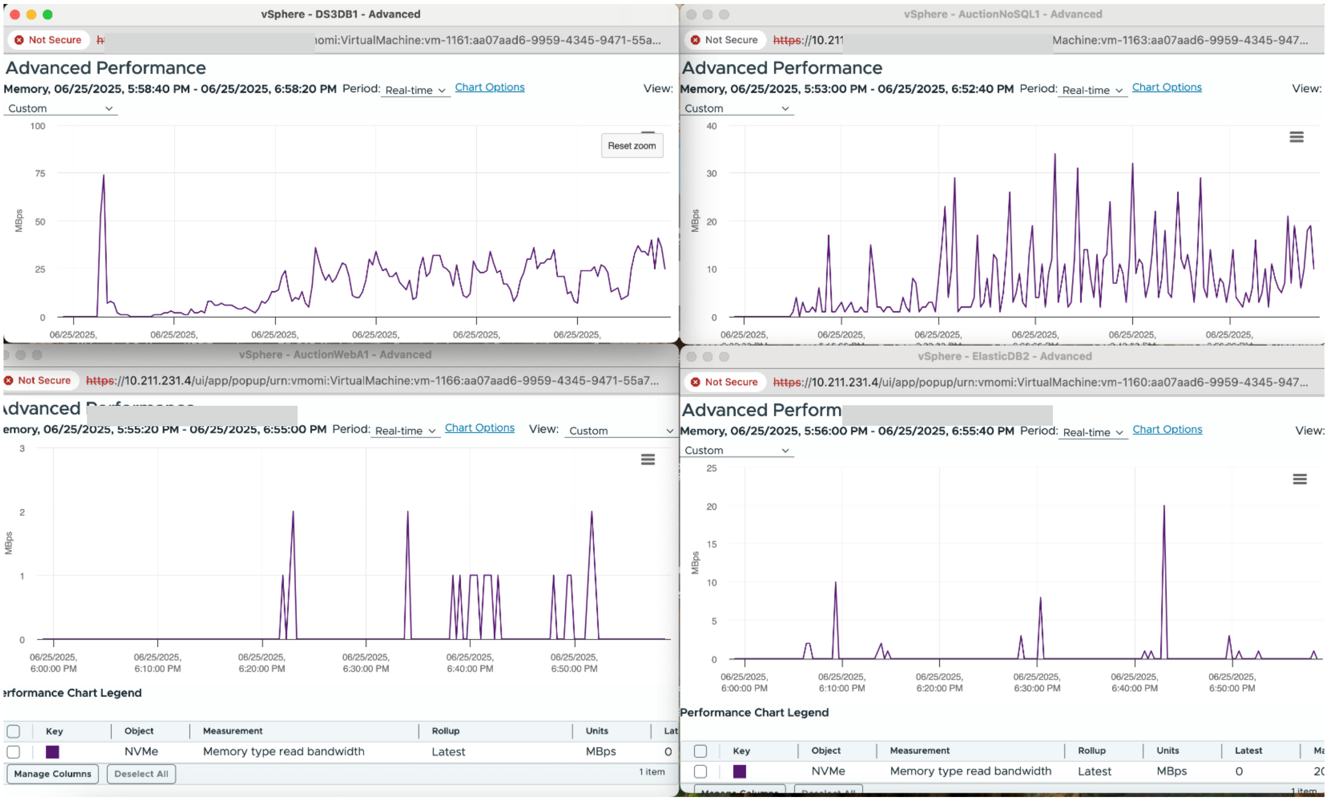
Performance Chart Legend

☐	Key	Object	Measurement	Rollup	Units	Latest	Maximum	Minimum	Average
☐	■	NVMe	Memory type read bandwidth	Latest	MBps	40	74	0	16.594

1 item

Figure 40 shows the read bandwidth for 4 VMs, as found by using the pop-out feature of a vCenter chart (click the three lines below the View dropdown menu). You can see the top two VMs have high read bandwidths. However, the bottom two VMs have minimal read bandwidths, which means they are not fetching much data from NVMe.

Figure 40. Advanced Performance memory view in vCenter with read bandwidth shown for the selected VMs



vSphere - DS3DB1 - Advanced

Not Secure ~~h~~[redacted]ıomi:VirtualMachine:vm-1161:aa07aad6-9959-4345-9471-55a...

Advanced Performance

Memory, 06/25/2025, 5:58:40 PM - 06/25/2025, 6:58:20 PM Period: Real-time Chart Options View: Custom

vSphere - AuctionNoSQL1 - Advanced

Not Secure ~~https~~://10.211[redacted]Machine:vm-1163:aa07aad6-9959-4345-947...

Advanced Performance

Memory, 06/25/2025, 5:53:00 PM - 06/25/2025, 6:52:40 PM Period: Real-time Chart Options View: Custom

vSphere - AuctionWebA1 - Advanced

Not Secure ~~https~~://10.211.231.4/ui/app/popup/urn:vmomi:VirtualMachine:vm-1166:aa07aad6-9959-4345-9471-55a7...

ıdvanced P[redacted]

emory, 06/25/2025, 5:55:20 PM - 06/25/2025, 6:55:00 PM Period: Real-time Chart Options View: Custom

ɜrformance Chart Legend

☐	Key	Object	Measurement	Rollup	Units	Lat
☐	■	NVMe	Memory type read bandwidth	Latest	MBps	0

1 item

vSphere - ElasticDB2 - Advanced

Not Secure ~~https~~://10.211.231.4/ui/app/popup/urn:vmomi:VirtualMachine:vm-1160:aa07aad6-9959-4345-947...

Advanced Perform[redacted]

Memory, 06/25/2025, 5:56:00 PM - 06/25/2025, 6:55:40 PM Period: Real-time Chart Options View: Custom

Performance Chart Legend

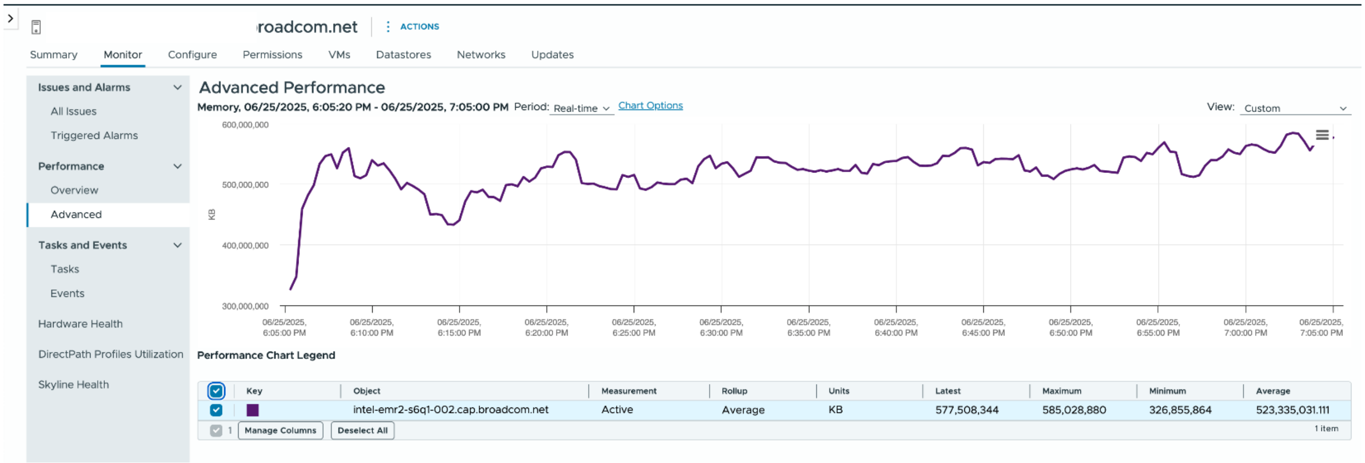
☐	Key	Object	Measurement	Rollup	Units	Latest	Ma
☐	■	NVMe	Memory type read bandwidth	Latest	MBps	0	2(

1 item

Active memory for host

Figure 41 shows the active memory for a host in vCenter.

Figure 41. Host active memory in vCenter shown by selecting **Monitor** > **Performance** > **Advanced** > **Chart Options**



Memory, 06/25/2025, 6:05:20 PM - 06/25/2025, 7:05:00 PM Period: Real-time

View: Custom

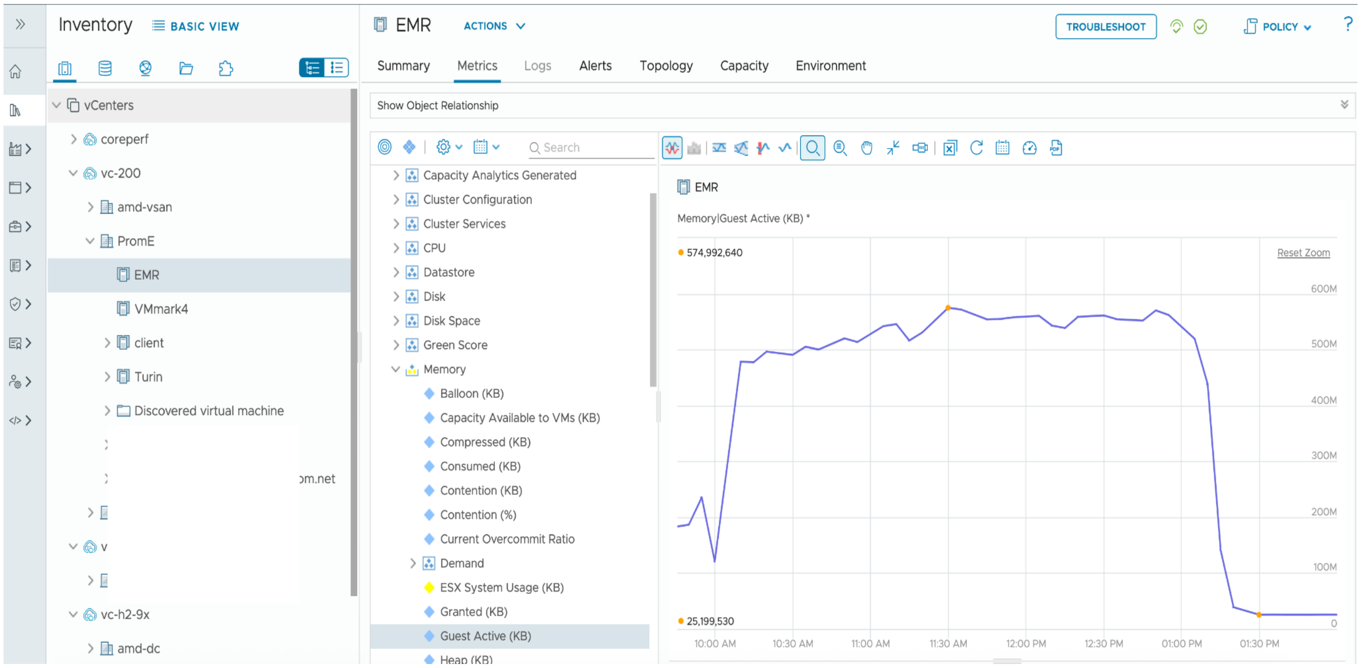
Performance Chart Legend

☑	Key	Object	Measurement	Rollup	Units	Latest	Maximum	Minimum	Average
☑	■	intel-emr2-s6q1-002.cap.broadcom.net	Active	Average	KB	577,508,344	585,028,880	326,855,864	523,335,031.111

1 item

You can also observe the same active memory using VCF Operations.

Figure 42. Active memory as seen in VCF Operations



Other recommendations

We also recommend you:

- Make sure there is sufficient CPU headroom to accommodate the CPU cost of memory tiering.
- Make sure host CPU utilization on a non-tiered host in the cluster does not exceed 75%. If it does, the efficiency of Memory Tiering might be reduced.
- Do not use "monster" VMs—those larger than 32 vCPUs and 512GB DRAM—with Memory Tiering.

Conclusion

Memory tiering, an important enhancement in VCF 9.0, takes advantage of adding NVMe SSDs to provide more memory capacity at a significantly lower cost than using only DRAM. This feature intelligently places frequently accessed data on high-speed DRAM and places less frequently accessed data to NVMe SSD. This strategy offers a solution to the problem of rising memory costs in datacenters by optimizing the total cost of ownership (TCO) for servers and for workloads bounded by memory capacity.

We demonstrated the performance of memory tiering by measuring it with several benchmarks and workloads. We saw consistent good performance, including a 2x increase in VM density and datacenter TCO savings of up to 40%. We also showed that memory tiering released extra host CPU capacity that would otherwise go unused in memory-capacity constrained scenarios.

Additionally, in NVMe-tiered settings, vMotion operations—a crucial part of virtualized environments—were barely affected by migrations, with VM downtime continuously staying below 1 second.

We found that you can ensure optimal performance by monitoring active memory. Ideally, keep it below 50% of DRAM. In addition, you should keep NVMe tier device latency below 200 microseconds.

About the authors

Qasim Ali and **Todd Muirhead**, performance engineers in the VMware Cloud Foundation (VCF) division at Broadcom, are the lead authors of this paper. Qasim is the performance lead for the Memory Tiering feature, while Todd works with databases, servers, and storage. He is also the co-creator and maintainer of the DVD Store open-source benchmark.

Other authors include Sharon Weber, James Zubb, Sreekanth Setty, and Praveen Vegulla.

Acknowledgments

Special thanks to Rajesh Venkatasubramanian and others on the team who thoroughly reviewed this paper.