Implementing Memory Tiering over NVMe using VMware ESXi 9.0

Planning / Implementation

With each new generation of data center servers, the CPU core count continues to increase. An increase in core count requires more memory capacity to make higher bandwidth data available to each processor core. However, the expansion of memory capacity has not kept pace with the rapid advancement of CPUs, resulting in a scenario where CPU utilization is often low, yet memory demand remains high. Specifically, CPU utilization frequently falls below 50%, primarily due to memory constraints that starve the systems.

When sufficient memory is not available to supply data to the CPU, even the most advanced processors are unable to reach their full potentials, highlighting the critical need for balanced system design.

Looking at the cost of a server by component, memory DIMMs stand out as one of the single most expensive components, and customers increasingly want to boost their memory capacity without incurring significant additional costs. The high cost of DRAM, particularly in data centers and AI/ML applications, substantially contributes to the Total Cost of Ownership (TCO). Furthermore, the number of slots in memory controllers is failing to keep pace with the rising core counts in multi-core processors, leading to a scenario where processor cores are often starved for memory bandwidth, thereby diminishing the benefits of higher core counts.

To effectively manage and reduce TCO, it is essential to optimize memory utilization and explore cost-effective alternatives, such as memory tiering, which can help strike a balance between performance and cost.

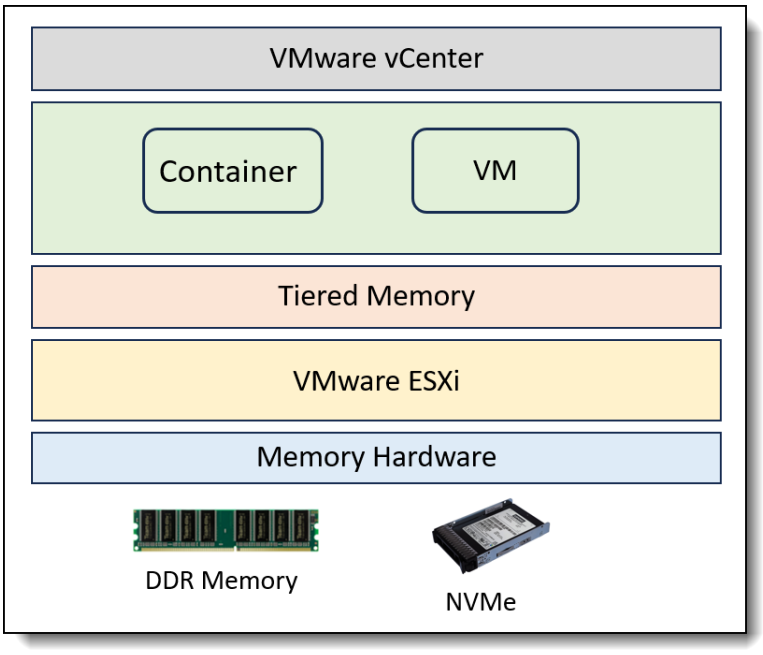

Figure 1. VMware memory hardware

The release of VMware Cloud Foundation (VCF) 9.0 marks a significant milestone for vSphere ESXi, as it introduces a new feature: Memory Tiering over NVMe. Initially previewed in vSphere ESXi 8.0 Update 3, this capability is now officially supported in vSphere ESXi 9.0. As illustrated in Figure 1, Memory Tiering over NVMe enables users to harness the power of NVMe devices as an additional layer of memory, substantially increasing memory capacity at a lower cost than traditional DRAM. This, in turn, facilitates host densification, leading to improved CPU utilization and enhanced overall system efficiency.

A crucial element of memory tiering is the incorporation of NVMe devices, renowned for their exceptional high-speed and low-latency performance. These devices function as a secondary layer to traditional DRAM, effectively augmenting system memory capacity. Notably, each NVMe SSD have a direct PCIe x4 connection, which yields significantly greater bandwidth and lower latency compared to SATA/SAS-based SSD solutions. The advantages of higher-capacity drives and accelerated speeds are evident in server SSD trends, which demonstrate a strong adoption of the PCIe interface and higher-capacity drives, underscoring the industry's shift towards faster and more efficient storage solutions.

VMware's memory management plays a vital role in the implementation of memory tiering, as the ESXi hypervisor employs intelligent memory allocation. By dynamically placing frequently accessed data in the faster DRAM and less frequently used pages in the NVMe tier, this approach leverages a tiered memory architecture to create an expanded memory pool that can accommodate diverse performance requirements. Notably, this tiering process is entirely transparent to the application, ensuring seamless compatibility with all types of workloads, regardless of their specific demands or characteristics.

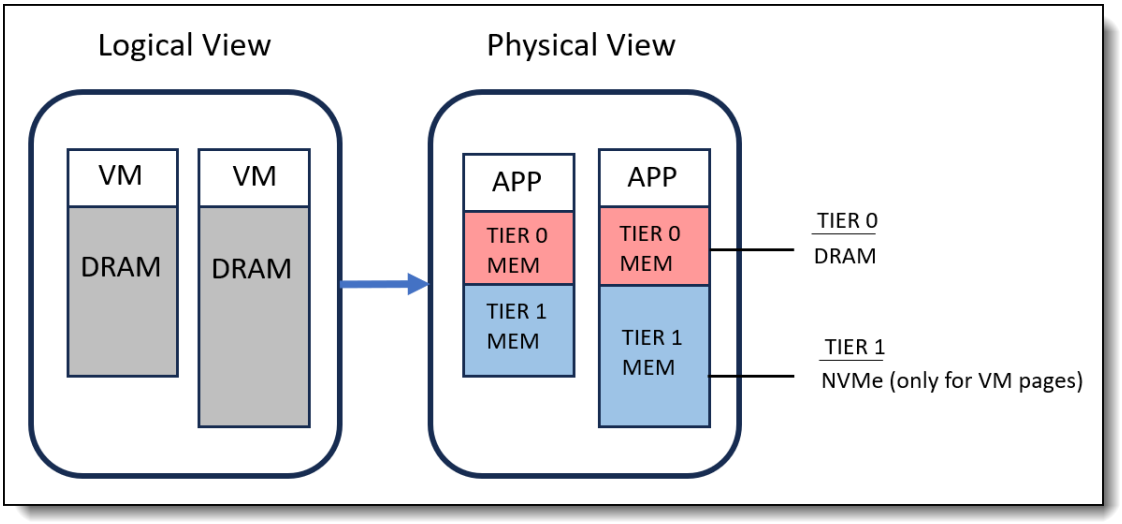

Figure 2. Memory tiering architecture

As shown in Figure 2, memory tiering integrates DRAM (TIER 0) with slower NVMe memory (TIER 1) to form a contiguous memory space. Through intelligent and proactive page placement, memory pages are dynamically allocated between the primary DRAM and secondary NVMe tiers. Specifically, memory pages from NVMe are utilized exclusively for VM memory allocation on the ESXi host, ensuring efficient use of resources.

Memory tiering enables customers to expand their memory footprint, increase workload capacity, and achieve better VM consolidation ratios. Following internal testing conducted by Broadcom at VMware, it was observed that memory tiering can yield significant benefits, including up to 40% savings in TCO for most workloads. Additionally, memory tiering can also unlock increased CPU utilization, with some workloads experiencing a boost of up to 25-30% more cores, thereby maximizing resource efficiency and optimizing system performance.

CXL (Compute Express Link) is a hardware interconnect standard that enables devices to connect to the CPU, memory, and peripherals. CXL-based memory is expected to provide larger and more scalable memory capabilities that can serve as a Tier 1 device in the ESXi memory tiering framework, offering further improvements in performance and greater memory bandwidth. VMware has been collaborating with CXL partners to enable this technology in future ESXi release. In this white paper, we will focus on memory tiering over NVMe devices.

Memory Tiering Setup

In this section, we outline the steps to configure and enable memory tiering on a VMware ESXi 9.0 host. The test configuration used in this scenario is based on the ThinkSystem SD535 V3, with detailed specifications listed in the following table.

Table 1. ThinkSystem SR535 V3 server configuration

Component	Configuration
Server	ThinkSystem SD535 V3 Server
CPU	1x AMD EPYC 9965 192-Core Processor
Memory	2x 16GB DDR5 6400MHz RDIMM
SSD	480GB SATA SSD
NVMe	Samsung PM9D3a 1.92TB NVMe PCIe 5.0 x4 HS SSD
Host OS	ESXi 9.0 Custom Image for Lenovo ThinkSystem

To use the memory tiering feature, the environment must meet the following

requirements. For a more detailed overview of the vSphere requirements for memory tiering, please consult the Broadcom documentation.

- ESXi 9.0 and later
- vCenter 9.0 and later
- The NVMe device used for memory tiering must be of a similar class to a vSAN cache device
  For more information about the supported NVMe devices, see the Broadcom Compatibility Guide https://compatibilityguide.broadcom.com/ and in the Global Search field enter vSAN SSD. On the vSAN SSD page, apply these additional filters, as below Table 2 shows.

  Table 2. Search criteria of supported NVMe devices

Search Criteria	Value
Global Search	vSAN SSD
Device Type	NVMe
Endurance Class	Class D >= 7300 TBW
Performance Class	Class F: 100,000-349,999 Writes Per Second or Class G: 350,000 Writes Per Second

- NVMe device must be installed locally on the ESXi host, and can't be over Ethernet. The tier partition cannot be larger than 4TB.
- Memory Tiering supports the use of NVMe devices behind a RAID controller. To ensure device resiliency in case of a single device failure, determine if RAID can be a good consideration.

In this white paper, we will provide a step-by-step demonstration of how to configure and activate memory tiering using both the vSphere Client UI and ESXCLI commands. As outlined in the table below, we will walk through each configuration step, highlighting the specific actions performed using both the vSphere Client and ESXCLI.

Note: The Configuring NVMe devices step can only be performed using the ESXCLI command.

Table 3. Memory Tiering Setup

Operations	vSphere Client	ESXCLI command
1. Identify NVMe devices	Yes	Yes
2. Configuring NVMe devices	No	Yes
3. Configuring Memory Tiering	Yes	Yes
4. Verify Memory Tiering	Yes	Yes

These operations are discusses in the next 4 sections.

Identify NVMe devices

This section shows identifying the NVMe device configured for memory tiering.

If you're using ESXCLI, the steps are as follows:

1. Install ESXi 9.0 on the system and access the Direct Console User Interface (DCUI) by pressing the ALT+F2 key combination.
2. List the NVMe devices on the host and identify the NVMe device name using the following commands, as displayed in Figure 3.

   ```
   esxcli storage core adapter device list
   ```

   In this example, the NVMe device name is determined to be "eui.3739353057100203002538470000001", with a corresponding device path of "/vmfs/devices/disks/eui.3739353057100203002538470000001".

   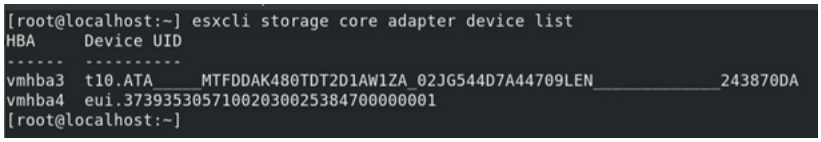

   Figure 3. NVMe device name

3. If the NVMe device already has an existing partition, use the ESXCLI partedUtil utility to delete the partition, as illustrated in Figure 4. This step is necessary to ensure that the NVMe device is properly prepared for use with memory tiering.

   ```
   partedUtil get /vmfs/devices/disks/eui.3739353057100203002538470000001
   partedUtil delete /vmfs/devices/disks/eui.3739353057100203002538470000001 1
   partedUtil get /vmfs/devices/disks/eui.3739353057100203002538470000001
   ```

   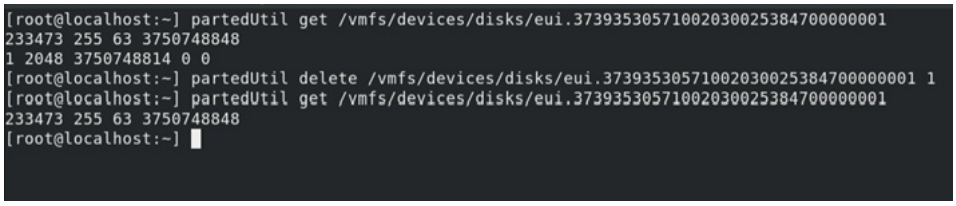

   Figure 4. partedUtil utility to delete partition

If you're using the vSphere client, the steps are as follows:

1. Log in to VMware vCenter and add the ESXi 9.0 host to the data center.
2. Select the ESXi host and navigate to the Configure tab, then click on Storage and select Storage Devices.
3. From the list of available devices, select the NVMe device and note the device name, as illustrated in Figure 5. This step is crucial in identifying the correct device for configuration.

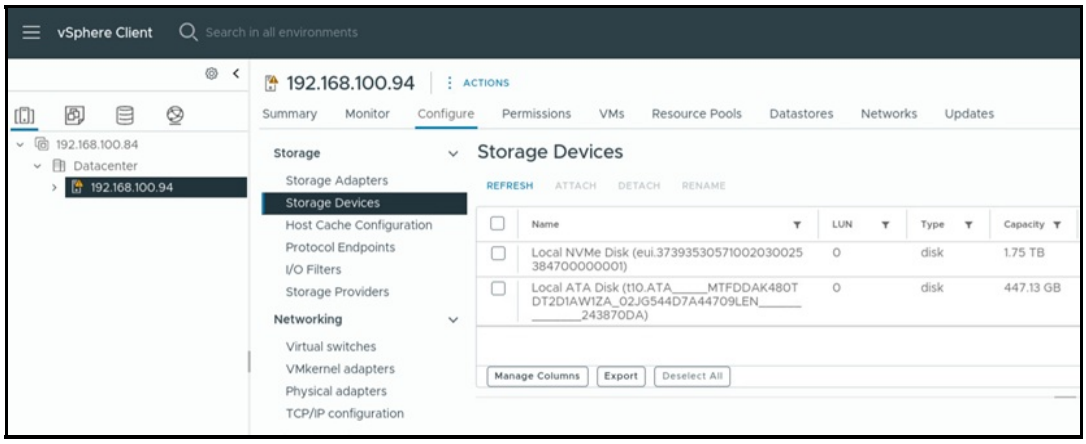

Figure 5. NVMe device in vSphere client

4. If an existing partition is present on the NVMe device, delete it by selecting the "ERASE PARTITIONS" option, which will remove all partitions from the selected disk, as illustrated in Figures 6 and 7. This step is necessary to ensure that the NVMe device is properly prepared for use with memory tiering.

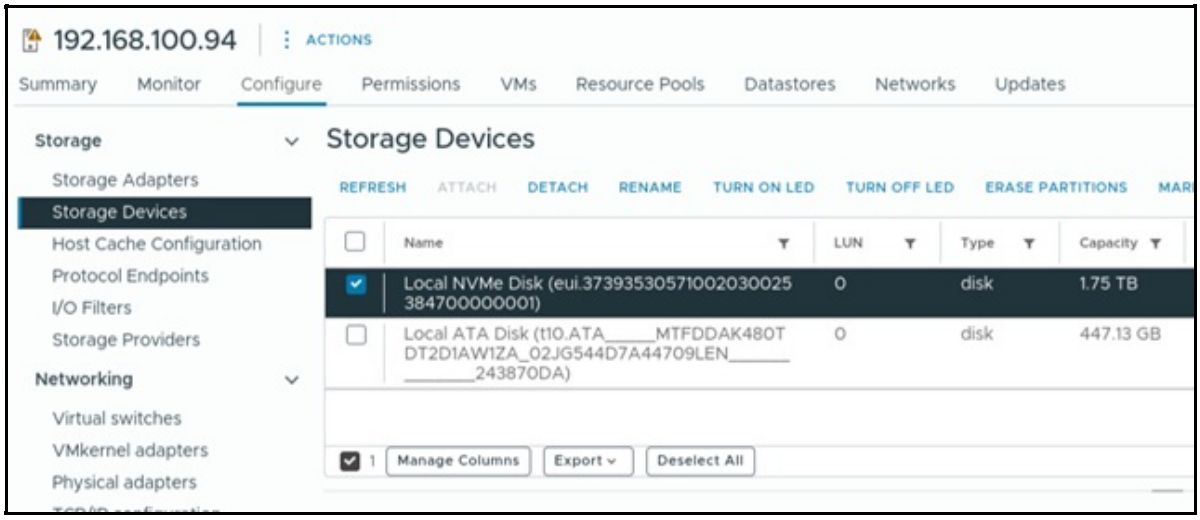

Figure 6. Erase NVMe partition

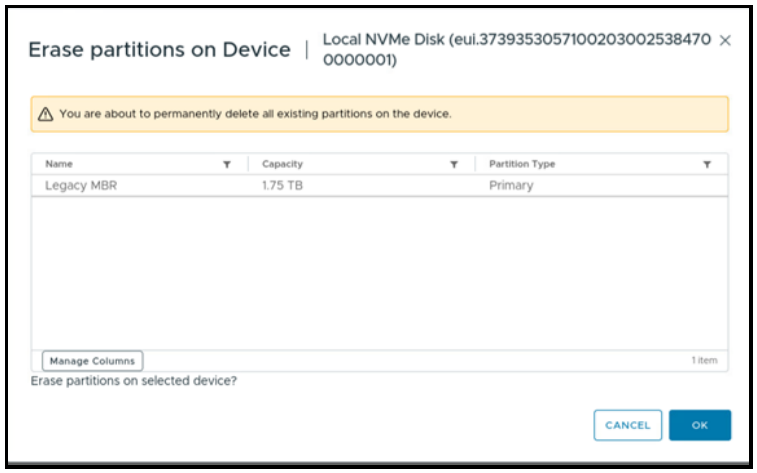

Figure 7. Confirm to erase NVMe partition

Configuring NVMe devices

This section shows creating a tier partition on the NVMe device

This step can only be performed using ESXCLI, as follows:

1. Create a single tier partition on the NVMe device and verify the tier partition was created, as shown in Figure 8.

```
esxcli system tierdevice list
esxcli system tierdevice create -d /vmfs/devices/disks/eui.37393530571002030025
384700000001
esxcli system tierdevice list
```

Figure 8. Create a tiered partition on NVMe device

Configuring Memory Tiering

This section shows activating the memory tiering feature for an ESXi host.

If you're using ESXCLI, the steps are as follows:

1. Configure the ESXi boot option "MemoryTiering" to activate the memory tiering feature, use the commands below:

```
esxcli system settings kernel set -s MemoryTiering -v TRUE
reboot
```

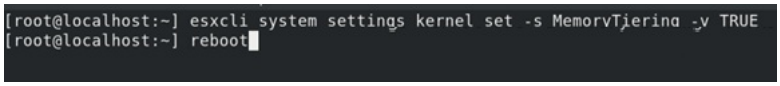

Figure 9. Kernel parameter

2. When enabling memory tiering over NVMe, there are two optional host advanced settings that can be configured: Mem.TierNvmePct and Mem.EncryptTierNvme, as illustrated in Figure 10. After modifying these host advanced settings, it is necessary to reboot the host in order for the changes to take effect.

```
esxcli system settings advanced set -i 100 -o /Mem/TierNvmePct
esxcli system settings advanced set -i 1 -o /Mem/EncryptTierNvme
```

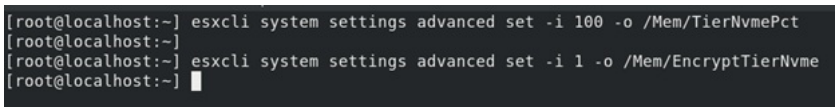

Figure 10. ESXCLI command to change advanced settings

The two optional host advanced settings for memory tiering over NVMe are as follows:

- **Mem.TierNvmePct**: This setting determines the amount of NVMe to be used as tiered memory, expressed as a percentage of the DRAM capacity. The default value is 100, which means that the host is configured to use a 1:1 DRAM:NVMe ratio. For example, a host with 1TB of DRAM would use 1TB of NVMe as tiered memory.

```
Path: /Mem/TierNvmePct
Type: integer
Int Value: 100
Default Int Value: 100
Min Value: 1
Max Value: 400
String Value:
Default String Value:
Valid Characters:
Description: NVMe size for memtiering (in percentage of DRAM)
Host Specific: false
Impact: reboot
```

Figure 11. Advanced setting Mem.TierNvemPct

- **Mem.EncryptTierNvme**: This setting enables the encryption of VM memory pages when tiering out to NVMe storage, providing an additional layer of data security. The default value is 0, indicating that encryption is disabled by default.

```
Path: /Mem/EncryptTierNvme
Type: integer
Int Value: 0
Default Int Value: 0
Min Value: 0
Max Value: 1
String Value:
Default String Value:
Valid Characters:
Description: Encrypt when tiering out to NVMe storage.
Host Specific: false
Impact: none
```

Figure 12. Advanced setting Mem.EncryptTierNvme

To enable memory tiering using the vSphere Client, follow these steps:

1. Select the host in the vSphere Client and navigate to Configure -> System -> Advanced System Settings.
2. Click the EDIT button to modify the settings.
3. Search for the key "VMkernel.Boot.memoryTiering" and set its value to "true" to enable memory tiering.
4. For the two optional host advanced settings, search for the keys "Mem.TierNVMePct" and "Mem.EncryptTierNvme" and enter the desired values.
5. Verify the changes, as shown in Figure 13.

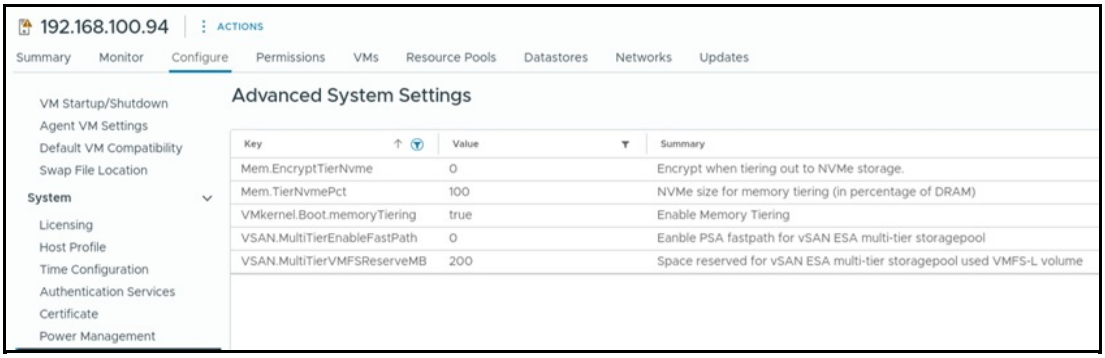

Figure 13. Host advanced setting in vSphere client

6. Reboot the host to apply the changes

Verify Memory Tiering

This section shows verifying whether a host is configured for memory tiering.

If you're using ESXCLI, the steps are as follows:

1. Issue the command shown in Figure 14. As shown, the memory size has increased to approximately 64GB, which is roughly twice the size of the original memory. This significant expansion in memory capacity is a direct result of enabling memory tiering.

```
esxcli hardware memory get
```

```
[root@localhost:~] esxcli hardware memory get
   Physical Memory: 67821203456 Bytes
   Reliable Memory: 0 Bytes
   NUMA Node Count: 1
[root@localhost:~]
```

Figure 14. Memory size after enable memory tiering

2. Use the command displayed in Figure 15 to view detailed information about Tier 0 and Tier 1 memory, This command provides insights into the memory hierarchy, where Tier 0 represents the amount of DRAM present in the system, and Tier 1 represents the amount of NVMe device being utilized as tiered memory, effectively expanding the system's memory capacity.

```
vsish -e get /memory/tiers/0/info
vsish -e get /memory/tiers/1/info
```

```
[root@localhost:~] vsish -e get /memory/tiers/0/info
Memory tier information {
   Memory tier type:0
   A more descriptive name of this memory tier:DRAM
   Memory tier flags:1
   Memory tier Internal flags:1
   Memory tier size (bytes):33909784576
   Memory tier free (bytes):27043840000
}
[root@localhost:~] vsish -e get /memory/tiers/1/info
Memory tier information {
   Memory tier type:2
   A more descriptive name of this memory tier:NVMe
   Memory tier flags:8
   Memory tier Internal flags:0
   Memory tier size (bytes):33909780480
   Memory tier free (bytes):33909780480
}
[root@localhost:~]
```

Figure 15. Tier 0 and Tier 1 memory size

If you're using the vSphere client, the steps are as follows:

1. In the vSphere Client, navigate to the host and select Configure -> Hardware -> Overview, as shown in Figure 16.
2. In the Memory table, verify that the Memory Tiering column displays "Software", indicating that memory tiering is enabled.
3. Additionally, verify that the table shows that both Tier 0 and Tier 1 have approximately 32GB of memory allocated, providing a clear overview of the memory configuration.

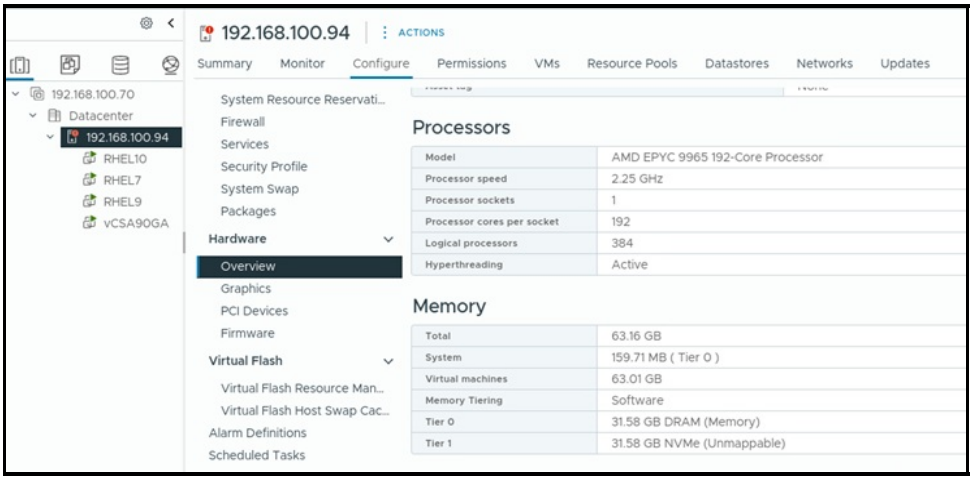

Figure 16. Memory information in vSphere client

VMware Memory Usage Utility

VMware provides the memstats utility which includes a new report type called vmtier-stats. This report offers a range of metrics that can be used to monitor the amount of VM memory that has been offloaded to Tier 1 memory compared to Tier 0 memory.

The table below lists some key metrics of interest, which can be used to analyze and optimize memory tiering performance.

Table 4. vmtier-stats report metrics

Report header label	Description
name	virtual machine group name (VM Cartel ID)
memSize	VM configured memory size
active	Total guest memory active
tier0Consumed	Total memory consumed in Tier 0 (DRAM)
tier1Consumed	Total memory consumed in Tier 1 (NVMe Tiering)

As illustrated in Figure 17, there are 4 VMs running on an ESXi server, which is equipped with 32GB of DRAM and has NVMe tiering enabled with a 100% ratio. The memory configuration shows that the total configured memory is approximately 61GB, with around 27GB of active memory being utilized. Notably, the Tier 0 memory (DRAM) is around 28GB, indicating that not all of the physical DRAM is being fully consumed. Meanwhile, the Tier 1 memory (NVMe) is approximately 18GB, demonstrating the effective use of NVMe storage as a secondary tier of memory.

```
[root@localhost:~] memstats -r vmtier-stats -u mb -s name:memSize:active:tier0Consumed:tier1Consumed

VIRTUAL MACHINE MEMORY TIER STATS: Thu Jun 26 12:50:47 2025
 ------------------------------------------
   Start Group ID   : 0
   No. of levels    : 12
   Unit             : MB
   Selected columns : name:memSize:active:tier0Consumed:tier1Consumed

-------------------------------------------------------------------
          name    memSize     active tier0Consumed tier1Consumed
-------------------------------------------------------------------
     vm.528332      20480       3792          7447          8585
     vm.528630       8192        276           884           289
     vm.528865      16384      12713         11780          3038
     vm.528996      16384      10754          8326          6462
-------------------------------------------------------------------
         Total      61440      27533         28436         18373
-------------------------------------------------------------------
[root@localhost:~]
```

Figure 17. memstats utility

To optimize memory performance and minimize latency, it is recommended that the amount of NVMe configured as tiered memory does not exceed the total amount of DRAM on the ESXi host. Furthermore, to avoid latency issues, the total amount of active virtual machine memory should fit within the total DRAM available to virtual machines on the host. Additionally, to ensure optimal performance, it is suggested that the active memory not exceed 50% of the DRAM size, thereby preventing potential performance degradation.

Summary

The memory tiering over NVMe feature in vSphere ESXi 9.0 is a game-changing capability that offers substantial benefits in terms of expanded memory capacity, optimized performance, cost efficiency, and enhanced workload consolidation. By elevating memory virtualization to the next level, memory tiering enables customers to reduce infrastructure costs and improve resource utilization without compromising performance.

By following the step-by-step guidelines outlined in this white paper, administrators can successfully configure and leverage this feature to unlock the full potential of their vSphere ESXi 9.0 hosts and maximize the value of their infrastructure investments.

References

For more information, see these resources:

- Advanced Memory Tiering Now Available with VMware Cloud Foundation 9.0
  https://blogs.vmware.com/cloud-foundation/2025/06/19/advanced-memory-tiering-now-available
- Memory Tiering over NVMe
  https://techdocs.broadcom.com/us/en/vmware-cis/vsphere/vsphere/9-0/vsphere-resource-management/memory-tiering-over-nvme.html
- Memory Tiering Performance - VMware Cloud Foundation 9.0
  https://www.vmware.com/docs/memtier-vcf9-perf

Author

Alpus Chen is an OS Engineer at the Lenovo Infrastructure Solutions Group in Taipei, Taiwan. As a specialist in Linux and VMware technical support for several years, he is interested in operating system operation and recently focuses on VMware OS.

Related product families

Product families related to this document are the following:

- Drives
- Memory
- VMware vSphere

Notices

Lenovo may not offer the products, services, or features discussed in this document in all countries. Consult your local Lenovo representative for information on the products and services currently available in your area. Any reference to a Lenovo product, program, or service is not intended to state or imply that only that Lenovo product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any Lenovo intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any other product, program, or service. Lenovo may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

Lenovo (United States), Inc.
8001 Development Drive
Morrisville, NC 27560
U.S.A.
Attention: Lenovo Director of Licensing

LENOVO PROVIDES THIS PUBLICATION ”AS IS” WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some jurisdictions do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. Lenovo may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

The products described in this document are not intended for use in implantation or other life support applications where malfunction may result in injury or death to persons. The information contained in this document does not affect or change Lenovo product specifications or warranties. Nothing in this document shall operate as an express or implied license or indemnity under the intellectual property rights of Lenovo or third parties. All information contained in this document was obtained in specific environments and is presented as an illustration. The result obtained in other operating environments may vary. Lenovo may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Any references in this publication to non-Lenovo Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this Lenovo product, and use of those Web sites is at your own risk. Any performance data contained herein was determined in a controlled environment. Therefore, the result obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

This document, LP2288, was created or updated on August 29, 2025.

Send us your comments in one of the following ways:

- Use the online Contact us review form found at:
  https://lenovopress.lenovo.com/LP2288
- Send your comments in an e-mail to:
  comments@lenovopress.com

This document is available online at https://lenovopress.lenovo.com/LP2288.

Trademarks

Lenovo and the Lenovo logo are trademarks or registered trademarks of Lenovo in the United States, other countries, or both. A current list of Lenovo trademarks is available on the Web at https://www.lenovo.com/us/en/legal/copytrade/.

The following terms are trademarks of Lenovo in the United States, other countries, or both:
Lenovo®
ThinkSystem®

The following terms are trademarks of other companies:

AMD and AMD EPYC™ are trademarks of Advanced Micro Devices, Inc.

Other company, product, or service names may be trademarks or service marks of others.