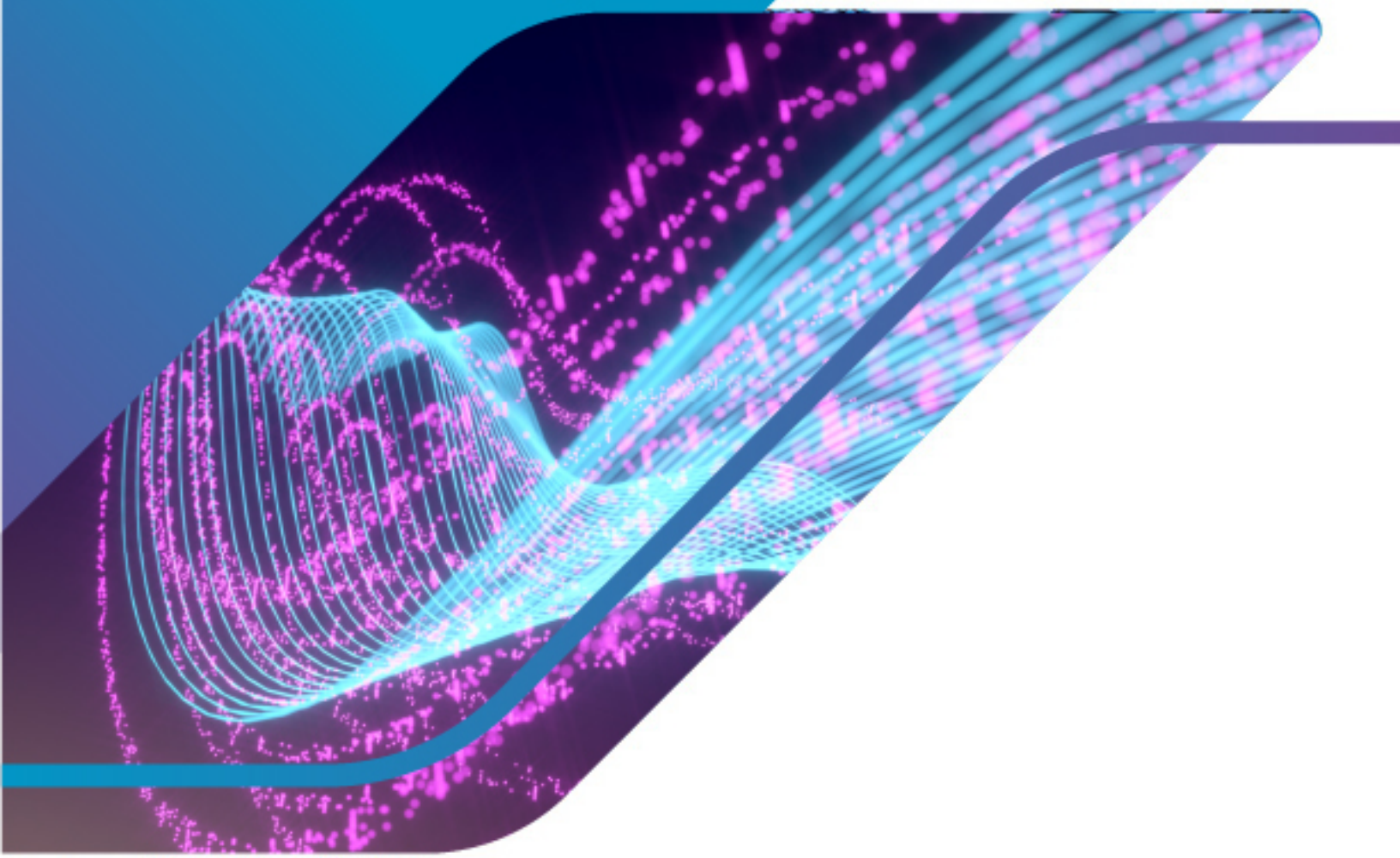
PERFORMANCE TECHNICAL WHITE PAPER
January 2025



Performance Tuning for Latency-Sensitive Workloads

VMware vSphere 8

vmware® by Broadcom

Table of contents

Overview

This guide documents Broadcom's best practices, considerations, and recommendations for running low-latency applications in vSphere 8.x, which has been optimized to support these types of applications.

WARNING: We strongly recommend you change vSphere latency tunings in a non-production environment and evaluate these suggestions in the context of your particular applications or workloads. After thorough testing, use only those tunings that help meet specific key performance indicators (KPIs).

Introduction

VMware vSphere® provides a high-performance and competitive platform that effectively runs many tier 1 application workloads in VMs. Developers tuned vSphere to drive high I/O throughput efficiently by utilizing fewer CPU cycles and conserving power, as required by a wide range of workloads.

However, minimizing compute and I/O latency is necessary for many applications, even at the expense of higher CPU utilization and greater power consumption. The default platform configuration for vSphere with workload consolidation does not meet these applications' performance requirements.

Some real-time or latency-sensitive applications include media and entertainment streaming platforms, financial services market data processing, and real-time automation control systems. This class of application is extremely sensitive to CPU contention due to the need for deterministic or constant access to the physical processors and requires very low-latency network end-to-end transactions.

With the knowledge in this whitepaper, you can optimize vSphere hosts and VMs to support these types of workloads.

Note: The exact benefits and effects of each of these configuration choices depend on the specific applications and workloads you're using, so we strongly recommend experimenting with the different configuration options for your workloads before deploying them in a production environment.

Baseline requirements

Developers are continuously optimizing vSphere hardware and software, so we recommend using the most current versions of the following:

- Processor generations
- Supported BIOS and microcode
- vSphere 8.0 U3 (or newer) or at least 7.0 U3
- Virtual Hardware 21 (or newer)
- VM Tools 12.4.5 (or newer)

Because of the complexity of supporting low-latency applications on vSphere, we recommend using as few vSphere overlays as possible (for example, NSX and VSAN). Each of these services consumes host CPU cycles and can potentially increase latency.

Host considerations

At the foundation of a platform to support latency-sensitive workloads is a modern server with reasonably current processors. You should consider the core count and clock frequency. Higher clock frequencies generally benefit workloads with a small number of threads operating at high utilization, while applications consolidated or with a high number of threads operating in parallel benefit from larger core counts. Your application threading model and how you plan to deploy those applications will have an impact on processor selection.

Change the BIOS settings on the physical server where ESX is installed

The ESX host's BIOS has a significant performance impact on how the vSphere scheduler operates and services VMs hosting latency-sensitive applications.

Modern processors balance power consumption and performance by changing clock frequencies within parts of the processor package based on utilization, the thermal profile, and c-states. These automatic changes, however, can impact latency. Below are some physical server BIOS setting changes that typically help solve this problem, but you should also check with your server manufacturer's documentation for guidelines on optimizing their BIOS for virtualization and low latency.

Notes: The settings below retain turbo boost but use all core clock frequencies.

- **Power Management:** High Performance
- **Hyper-Threading:** Enabled
- **Turbo Boost:** Enabled
- **Energy Efficient Turbo:** Disabled
- **P States:** Disabled
- **C States:** Disabled
- **C1E Enhanced Mode:** Disabled
- **QPI/UPI Power Management:** Disabled
- **Node Interleaving:** Disabled

Disable EVC

vSphere EVC mode can mask processor instructions. Be sure to use the latest supported baseline for your processor (review Broadcom KB 313545 [1]) or disable EVC.

Carefully schedule vMotion and DRS

The published vMotion stun time is 1 second, although it often completes much faster than that with current optimizations. However, this cutover time frame is generally too long for latency-sensitive applications. As a result, only use vMotion and DRS within the proper change windows, and when you can move the application stack with this tolerance.

Enable or disable other advanced settings

The following sections describe our recommendations for other advanced settings.

Disable action affinity

Disable action affinity by setting `Numa.LocalityWeightActionAffinity` to `0` as described in Broadcom KB 314059 [2].

Open ring buffers

If the network streams are UDP-based, you may need to open the physical NIC rings to ensure no packets are lost:

```
esxcli network nic ring current set -n vmnicX -r 2048 -t 1024
```

Enable SplitRX

SplitRx mode uses multiple physical CPUs to process network packets received in a single network queue by default. This feature can significantly improve network performance for certain workloads, including multiple VMs on one ESX host, all receiving multicast traffic from the same source. If this feature is set to 0 (off), turn it on by changing it to 1 (on) [3].

```
NetSplitRxMode = "1"
```

Enable SplitTX

You configure SplitTX for a physical NIC on a host. When SplitTX is active, it allows the hypervisor to use two transmission threads for the pipeline: one for the virtual NIC backend and another for the physical NIC. When SplitTX is enabled, VM traffic using the vNIC connected to the pNIC will benefit from increased performance for packet egress. Since the configuration is host-based, you must use the following command to activate this feature for a specific vmnic:

```
vsish –e set /net/pNics/vmnicX/sched/txMode 1
```

Replace X with the network interface number.

Disable queue pairing

Queue pairing, a feature that some physical NICs support, tells the ESX uplink layer that the receive thread will also process the completion of transmitted packets on a paired transmit queue. In transmit-heavy media and entertainment workloads, this feature can lead to delays in processing transmit completions, which in turn can exhaust the transmit ring of the vNIC, causing the vNIC driver in the guest OS to drop packets.

To make room in the vNIC's transmit ring for more packets, you can disable this feature on an ESX host for all pNICs, which will create a separate thread for processing transmit completions for the pNICs.

The ESX command to disable queue pairing is:

```
esxcli system settings advanced set -o /Net/NetNetqRxQueueFeatPairEnable -i 0
```

For this to take effect, reboot the ESX host. Carefully consider disabling queue pairing, as it increases CPU utilization with an additional thread to handle transmit completions, thereby impacting the available CPU resources for other workloads on the same host.

VM considerations

Rightsize VMs

To best support low latency workloads, appropriately size a VM's vCPUs with the proper vTopolgy (see below). Also correctly size the VMs to the physical server that hosts them. This is critical to:

- Ensure the VM achieves the best performance.
- Reduce the hypervisor contention for the work it does on behalf of the VMs.

Simple rules:

- Don't size a VM larger than a physical NUMA node unless necessary. Keeping all vCPUs and memory within a physical NUMA node provides the best performance possible and reduces memory latency effects. When a VM is split across physical NUMA nodes, this causes memory to be remotely accessed, unless the application is already optimized for NUMA.
- Make the VM as small as possible, ensuring it is sized only for the performance needed now and soon. You can easily add resources to a VM when necessary.
- Leave resources for host services and I/O completions. In addition to hypervisor services, each ESX host has a number of worlds doing work on behalf of the VM. These include worlds receiving and transmitting network packets prior to the network traffic being passed in/out of the VM. To ensure you reduce scheduling contention on these worlds, and ensure low latency packet processing, you must save a number of physical cores (pCores) for these worlds and not allocate them to VMs. **We recommend that 4–8 pCores be excluded from the total host count.** There is no mechanism to protect these system worlds from contention other than rightsizing the VM and managing the VM population on the host.

Note: vSphere 7.0 and later versions don't include `sched.cpu.latencySensitivity.sysContexts`.

Use a current virtual hardware version

The virtual hardware version can mask processor-specific optimizations, so using the current version ensures the VM has access to all processor instructions. Refer to Broadcom KB 315655 [4] for a list of virtual hardware versions. We recommended using virtual hardware version 21 with vSphere 8.0 U3.

Leverage vTopology

Emulating the physical processor topology for VMs is critical for both execution and performance. ESX provides options to create virtual CPUs, virtual sockets, virtual cores per socket, virtual NUMA nodes, and virtual last-level caches (LLCs).

Before the new automation introduced in vSphere 8.0, the default configuration of a VM was one core per socket. This configuration created inefficiencies and needed manual tweaking to get optimal performance. The virtual topology presentation affects the guest OS behavior and its own scheduling.

By exposing the appropriate vTopology at various levels, vSphere 8 now applies the optimal vTopology to a VM's configuration. This happens the first time a new VM is powered on. You will still need to manually optimize a migrated VM.

For more information, see VM vCPU and vNUMA Rightsizing – Guidelines [5].

It is critical the vToplogy of the VM is aligned with the underlying hardware for latency-sensitive workloads.

Disable hot-add

A necessary performance optimization includes exposing the NUMA topology to the VM. This enables NUMA-aware applications in the guest to more effectively manage memory allocation based on which CPU the consuming thread is scheduled. Without vNUMA, application latencies might increase. Enabling vCPU hot-add results in the disabling of vNUMA, thereby exposing the guest OS to a less performant UMA topology.

Turn on latency sensitivity per VM

You can turn on latency sensitivity for each VM that requires it. This feature:

- **Gives exclusive access to physical resources to avoid resource contention due to sharing.**
  With exclusive physical CPU (pCPU) access, each vCPU entirely owns a specific pCPU; no other vCPUs and threads (including VMkernel I/O threads) are allowed to run on it. This achieves nearly zero ready time and no interruption from other VMkernel threads, improving response time and jitter under CPU contention.
- **Bypasses virtualization layers to eliminate the overhead of extra processing.**
  Once exclusive access to pCPUs is obtained, the feature allows the vCPUs to bypass the VMkernel's CPU scheduling layer and directly halt in the VM monitor (VMM) because there are no other contexts that need to be scheduled. This avoids the cost of running the CPU scheduler code and switching between the VMkernel and VMM, leading to much faster vCPU halt/wake-up operations.
- **Tunes virtualization layers to reduce the overhead.**
  When the VMXNET3 paravirtualized device is used for VNICs in the VM, VNIC interrupt coalescing and LRO support for the VNICs are automatically disabled to reduce response time and jitter.

For the best performance, follow these steps:

1. Set the CPU reservation: Go to **Edit Settings** → **Virtual Hardware** tab → **CPU** → **Reservation** → Enter the number of MHz to reserve and click **OK**. (Usually use the **base clock freq × vCPU count**.)
2. Set the memory reservation: Go to **Edit Settings** → **Virtual Hardware** tab → **Memory** → **Reservation** → Select **Reserve all guest memory (All locked)** and click **OK**.
3. Set latency sensitivity to high: **Edit Settings** → **VM Options** tab → **Advanced** → **Latency Sensitivity**. Select **High** and click **OK**.

Use VMXNET3

Be sure to select and use the VMXNET3 paravirtualized network driver within the guest operating systems. This driver has been optimized to support the best performance for guest operating systems and its use is considered a standard practice.

Balance Tx and Rx

If a vNIC requires high throughput, whether just high Rx side throughput or bi-directional, we recommend you configure `ethernetX.pnicFeatures=4` and `ethernetX.ctxPerDev=3` in the VM's VMX file to get the parallelism in the networking layer. From a bi-directional test with Netperf, we observed this gives a more balanced Tx-side and Rx-side bandwidth, roughly equally. These tunings work best with RSS enabled in the physical NIC.

Note: `ethernetX` corresponds to the vNIC being configured. Refer to KB 312057 [6] for details.

Exclusively affinitize the VM's Tx thread

Additionally, you can exclusively affinitize the VM's Tx thread by finding its GID process number.

Use esxtop, expand VM worlds, and look for `NetWorld-Dev-xxx-TX`:

1. Run:

```
vsish
```

2. Run:

```
set /sched/cpuClients/<WORLD_ID>/vcpuAffinities/<WORLD_ID> <"PCPU#">
```

   `<WORLD_ID>` is the process ID for the VM Tx thread and `<"PCPU#">` is the physical CPU number (for example: `"2"`) where you want to pin the thread outside of the vCPU count, but typically within the same physical NUMA node as the egress physical NIC.

3. Run:

```
set /sched/Vcpus/<WORLD_ID>/exclusiveAffinity 1
```

Using this latency sensitivity policy, the vCPUs are exclusively assigned to a physical CPU core. This command ensures the VM's Tx thread is also exclusively assigned to a physical CPU core.

Associate VMs with specified NUMA nodes

It may be useful to ensure the VM is aligned to the pNUMA node that the network adapter is electrically connected to. This can be done for the VM using:

```
numa.nodeAffinity = X
```

Refer to Associate Virtual Machines with Specified NUMA Nodes [7] for more information.

Use SR-IOV or DirectPath I/O if needed

SR-IOV and DirectPath I/O are technologies that leverage hardware support to allow VMs to directly access hardware devices. Consider these if VMXNET3 can't meet your performance requirements.

A VM with SR-IOV or DirectPath I/O can directly access the physical NIC instead of using a VMXNET3. While all these technologies can sustain high throughput of beyond 10Gbps, using SR-IOV or DirectPath I/O can additionally save CPU cycles in workloads with very high packet counts per second (for example, greater than 50,000 packets per second).

DirectPath I/O and SR-IOV have similar functionality, but you use them to accomplish different things:

- DirectPath I/O allows one VM to access one physical device.
- SR-IOV allows multiple VMs to share a single physical device.

Important: SR-IOV and DirectPath I/O do not support many virtualization features such as allowing VMs to share multiple NICs, memory overcommitment, vMotion, and network I/O control. Therefore, we recommend using these technologies only for workloads that may not require these features.

Enable other advanced settings

We recommend you enable three additional advanced settings. To do so:

1. Go to **Edit Settings** → **VM Options** tab → **Advanced** → **Configuration Parameters** → click **Edit Configuration** → Click on **Add Configuration Parameters**.
2. Add the following configuration parameters:

```
sched.cpu.affinity.exclusiveNoStats="TRUE"
monitor.forceEnableMPTI="TRUE"
timeTracker.lowLatency="TRUE"
```

Networking

We recommend that you almost always use the VMXNET3 paravirtualized network driver. Other options are mentioned above in this section: Use SR-IOV or DirectPath I/O if needed. In particular, you'd use these for a network load of over 50,000 packets per second and if VMXNET3 wasn't giving the performance you needed.

In addition to using VMXNET3, you can try adding one of the following technologies:

- Enhanced datapath to improve the speed of NFV workloads
- SmartNICs to offload network stack functions

Use enhanced datapath for NFV workloads

Enhanced datapath is a networking stack mode that provides superior network performance. It is primarily targeted for NFV workloads that require higher performance than regular workloads. Most workloads and platforms do not require these enhancements.

On an ESX host, configure a vSphere distributed switch in NSX with enhanced datapath. For details, refer to Enhanced Datapath [8].

Offload vSphere services to data processing units (DPUs), aka SmartNICs

vSphere 8 now supports the use of specific DPUs, also known as smartNICs, to offload vSphere services, including network stack functions. It enhances infrastructure services' performance with lower latency and higher throughput [9].

Using VMXNET3 UPTv2 mode, a VM's network traffic can be passed directly to the DPU and its network interface while still offering the flexibility of using DRS, HA, and other vSphere 8 virtualization features.

For details, refer to What is VMware vSphere Distributed Services Engine [10].

Guest OS and application (workload) tuning

Many operating systems come with different types of tunables. We recommend using the guest OS and/or application vendors' suggestions for low-latency tuning.

Photon OS tuning

Photon OS provides a secure Linux runtime environment and a real-time kernel called **linux-rt** to support low-latency workloads. This kernel is based on the Linux kernel **PREEMPT_RT** patch that turns Linux into a real-time operating system. In addition to the real-time kernel, Photon OS 3.0 supports several userspace packages such as **tuned**, **tuna**, and **stalld**. These are useful to configure the operating system for real-time workloads. The linux-rt kernel and the associated userspace packages together are referred to as Photon Real Time (aka Photon RT).

Operations guidance

There are a few ESX tools you can use to manage operations tasks:

- Configuration: **esxcli** and **vsish**
- Performance: **esxtop** and **net-stats**
- Management, logging, and diagnostics: VMware Cloud Foundation® Operations (aka **VCF Operations**)

Below we show some examples of commonly used commands.

Examples of commonly used commands

View network stats for physical NICs

Use the following **esxcli** command to view network stats for physical NICs:

```
esxcli network nic stats get -n vmnicX
```

View and/or set ring buffers for physical NICs

Use the following commands to check the:

1. Maximum Rx ring preset
2. Current Rx ring buffer size

```
esxcli network nic ring preset get -n vmnicN
esxcli network nic ring current get -n vmnicN
```

Note: Replace the **N** in the examples here with a number. In this example, it's the number of the vmnic.

Review the output and check the current Rx ring buffer size compared to the preset maximum Rx ring buffer size the NIC supports.

Use the following command to set the ring buffers:

```
esxcli network nic ring current set -n vmnicN -r N
```

You can also use `-t N` to change the transmit buffer if needed.

For more information, refer to Broadcom KB 341594 .

Use net-stats

For example:

```
net-stats -A -t WwqQi -i 2
```

1. |grep used to look at thread utilization and contention.
2. Check rxeps, txeps to identify drops.

List of esxcli network commands

The following table shows other helpful esxcli network commands.

network nic coalesce get	Get coalesce parameters
network nic coalesce set	Set coalesce parameters on a NIC
network nic cso get	Get checksum offload settings
network nic cso set	Set checksum offload settings on a nic
network nic ring current get	Get current RX/TX ring buffer parameters of a NIC
network nic ring current set	Set current RX/TX ring buffer parameters of a NIC
network nic ring preset get	Get preset RX/TX ring buffer parameters of a NIC
network nic stats get	Get NIC statistics for a given interface
network nic tso get	Get TCP segmentation offload settings
network nic tso set	Set TCP segmentation offload settings on a nic
network nic vlan stats get	List VLAN statistics for active VLANs on the NIC
network nic vlan stats set	Enable/disable VLAN statistics collection on the NIC
network port filter stats get	Filter statistics for a given port
network port stats get	Packet statistics for a given port
network sriovnic list	This command will list the SRIOV Enabled NICs (PFs) currently installed and loaded on the system
network vswitch standard portgroup list	List all of the port groups currently on the system
network nic list	List the Physical NICs currently installed and loaded on the system

Table 1. Useful esxcli network commands **Source:** Media and Entertainment Workloads on vSphere 6.7 [11]

Conclusion

This paper provides guidance for configuring the vSphere platform to host latency-sensitive applications and shares best practices that Broadcom and its partners have used to successfully enable this.

References

[1] Broadcom, "VMware EVC and CPU Compatibility FAQ," 12 November 2024. https://knowledge.broadcom.com/external/article/313545/vmware-evc-and-cpu-compatibility-faq.html.

[2] Broadcom, "NUMA nodes are heavily load imbalanced causing high contention for some virtual machines," 9 March 2024. https://knowledge.broadcom.com/external/article/314059/.

[3] Broadcom, "Performance Best Practices for VMware vSphere 8.0 Update 3," 2024. https://www.vmware.com/docs/vsphere-esxi-vcenter-server-80U3-performance-best-practices.

[4] Broadcom, "Virtual machine hardware versions," 13 November 2024. https://knowledge.broadcom.com/external/article/315655/.

[5] M. Achtemichuk, "Virtual Machine vCPU and vNUMA Rightsizing – Guidelines," 30 November 2022. https://blogs.vmware.com/performance/2017/03/virtual-machine-vcpu-and-vnuma-rightsizing-rules-of-thumb.html.

[6] Broadcom, "High virtual network throughput performance tuning recommendation when using 100G network interface cards," 2024 31 Jul. https://knowledge.broadcom.com/external/article?articleNumber=312057.

[7] Broadcom, "Associate Virtual Machines with Specified NUMA Nodes," 28 Aug 2023. https://docs.vmware.com/en/VMware-vSphere/8.0/vsphere-resource-management/GUID-A80A6337-7B99-48C8-B024-EE47E2366C1B.html.

[8] Broadcom, "Enhanced Datapath," 24 Jul 2022. https://docs.vmware.com/en/VMware-NSX/4.0/administration/GUID-668EB7EF-3E39-46C8-AF2F-43B7DAB6D42E.html.

[9] Broadcom, "vSphere on DPUs – Prepare your infrastructure to meet the demands of next-gen apps," Oct 11 2022. https://www.vmware.com/docs/vmw-ebook-vsphere-on-dpus.

[10] Broadcom, "What is VMware vSphere Distributed Services Engine?," 23 Jan 2025. https://techdocs.broadcom.com/us/en/vmware-cis/vsphere/vsphere/8-0/esxi-installation-and-setup-8-0/introducing-vmware-vsphere-distributed-services-engine-and-networking-acceleration-by-using-dpus-install.html#GUID-EC3CE886-63A9-4FF0-B79F-111BCB61038F-en.

[11] "Media and Entertainment Workloads on vSphere 6.7," 27 Mar 2019. https://www.vmware.com/docs/media-workloads-on-vsphere67-perf.

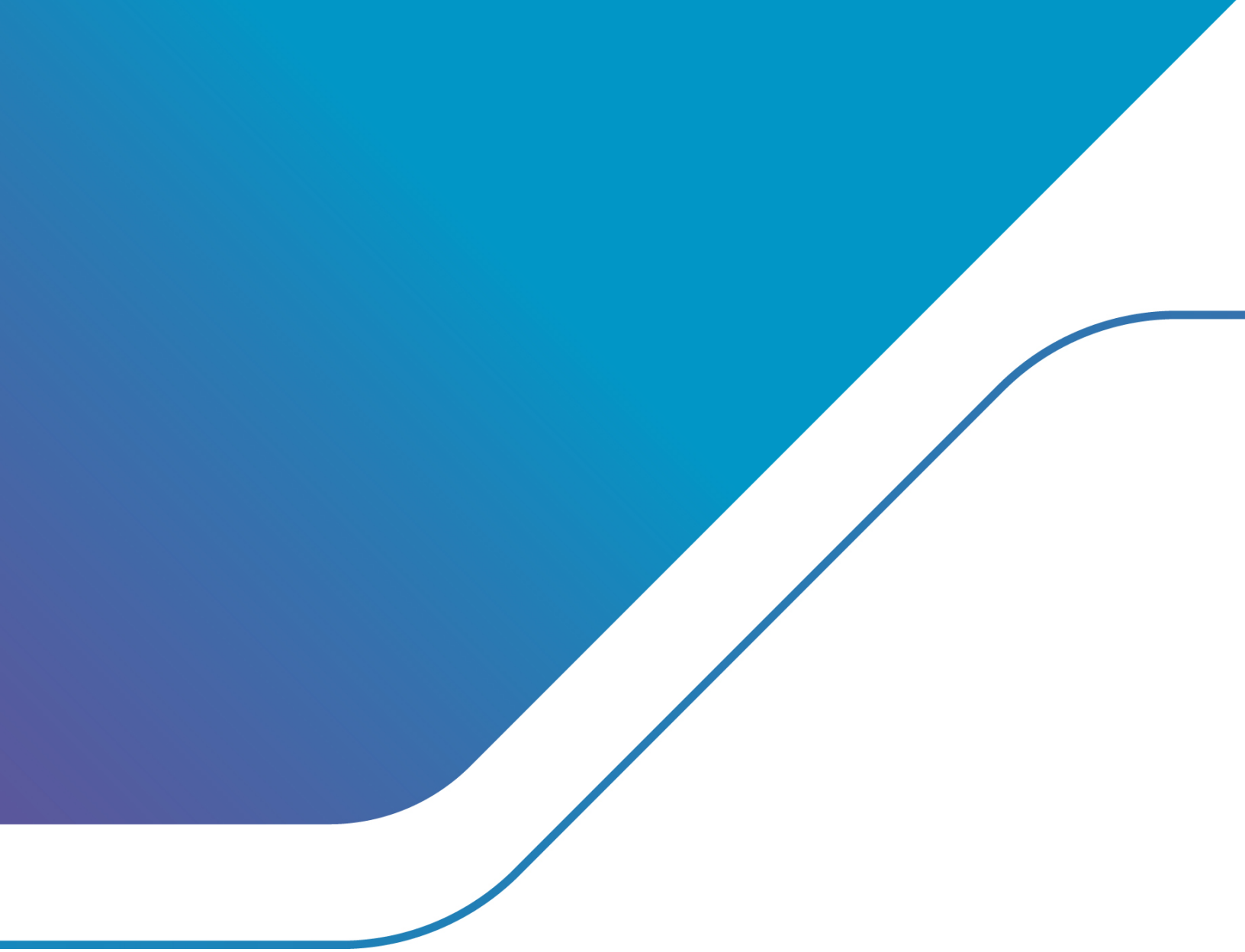
About the author

Mark Achtemichuk currently works as a master engineer within Broadcom's VMware Cloud Foundation (VCF) Performance Engineering team, focusing on education, benchmarking, customers, escalations, collateral, and performance architecture. He has also held various performance-focused fields, including specialist and technical marketing positions within VMware over the last 14+ years. Recognized as an industry expert, Mark holds a VMware Certified Design Expert (VCDX #50) certification, one of only 302 globally. He has worked on engagements with Fortune 500 companies, served as a technical editor for many books and publications, and is a sought-after speaker at numerous industry events. Mark is a blogger and has been recognized as a VMware vExpert from 2013 to 2024.

Acknowledgments

The following individuals from Broadcom contributed content or helped review this guide:

- **Yifan Hao**, software engineer, VMware Cloud Foundation (VCF) division
- **Samuel Kommu,** technology product management, VCF division
- **Julie Brodeur**, senior technical writer, VCF division, Performance Engineering team



vmware® by Broadcom