Technical Research Report

PROWESS

Accelerate Demanding Graphics and AI Workloads with GPUs and DPUs, Even in Virtualized Environments

Testing by Prowess Consulting determined that Dell™ PowerEdge™ servers configured with VMware vSphere® 8.0 Update 2 can make full use of hardware accelerators to enhance the performance of demanding workloads, even in a virtualized environment.

Executive Summary

It's no revelation to say that workloads are growing increasingly complex for enterprise and small and medium-size businesses (SMBs). AI and machine learning (ML) use cases demand high performance. 3D modeling, rendering, and advanced simulations are needed in architecture, financial services, healthcare, and manufacturing workflows. Media and entertainment organizations require complex rendering of video footage. Even database workloads need increasingly high compute power to keep up with modern needs in nearly every industry—from retail to finance to manufacturing and more.

One way to address this challenge is to employ hardware acceleration. For example, adding graphics processing units (GPUs) to data center servers offers the potential to boost performance for demanding workloads. In addition, data processing units (DPUs) can improve performance and reduce latency by offloading networking functions from the CPU. But are those performance gains realized when the workloads are running in virtualized environments, such as with VMware vSphere®?

In this study, commissioned by Dell Technologies, Prowess Consulting testing confirmed that GPUs and DPUs can provide significant performance increases for vSphere deployments.

Highlights

Compared to a non-accelerated deployment, adding GPUs and DPUs can significantly improve performance for demanding workloads in virtualized environments.[1]

>>

3,681% faster video rendering

36% higher iPerf® file-transfer performance

30% better CPU utilization

Study Motivation

More and more businesses today are staking their success on their ability to support complex workloads. Generative AI is increasingly used in coding, customer service, analytics, design, and even IT management and support functions. Engineers rely on complex 3D simulations for design work. Analysts use complex databases to extract actionable insights for sales, marketing, and finance. To support these scenarios, organizations require high-performance infrastructure. But rolling out high-end, costly servers to each user can be costly and cumbersome to manage and support.

Alternatively, organizations can rely on vSphere with virtual desktop infrastructure (VDI). This option lets IT administrators manage infrastructure and workloads centrally, while giving workers access to the applications they need through virtual workstations.

To fully support a virtualized environment with the required levels of performance for demanding workloads, organizations can install hardware accelerators on a host server running the VMware ESXi™ hypervisor. For example, businesses might choose to deploy vSphere on a supported and verified host, such as a Dell™ PowerEdge™ R760 server configured with GPU and DPU accelerators. End users could then run their compute-intensive workloads from virtual workstations, giving them access to both the software they need and the benefits provided by hardware acceleration on the host. Such an implementation can simplify deployment and management of hardware and software, and it can help reduce costs, because organization don't need to deploy multiple high-end physical workstations to meet the needs of their workers.

Accelerator Benefits

Businesses looking to accelerate graphics-dependent workloads can deploy GPUs on the servers that run the VMware ESXi hypervisor. Virtual desktops can then be configured to provide access to the GPU for enhanced performance of AI, complex design, 3D modeling, simulation, and other demanding workloads.

Similarly, organizations might want to enhance network performance in a virtualized environment by making use of a DPU on a vSphere server. A DPU can offload network and security functions, including IPsec encryption, Open vSwitch (OVS), and more, which can free up CPU cycles to process other tasks and improve overall system efficiency.

DPUs offer other benefits beyond performance gains. For example, DPUs can help reduce power usage by shrinking server footprint, especially in conjunction with vSphere virtualization. By deploying fewer server nodes running network and storage functions, organizations can reduce both capital expenditures (CapEx) and operating expenses (OpEx), leading to an improved return on investment (ROI) and reducing total cost of ownership (TCO). Although TCO calculations were beyond the scope of this Prowess Consulting study, a recent paper from The Next Platform highlights the TCO benefits that result from the performance gains and power savings provided by offloading IPsec encryption to a DPU.[2]

DPUs have the added benefit of strengthening the security footprint of the vSphere host server by enforcing security policies earlier, in the DPU hardware, before network traffic even reaches the hypervisor.[3]

Test Methodology

Prowess Consulting set out to investigate the performance benefits of two acceleration options when running in a VMware vSphere deployment:

- Using a GPU to accelerate the performance of a compute-intensive workload in a virtualized environment
- Using a DPU to accelerate network performance and reduce CPU utilization in a virtualized environment

Specifically, we tested a PowerEdge R760 server powered by 4th Gen Intel® Xeon® Scalable processors and configured with an NVIDIA® L40 GPU. We configured the virtual desktops with NVIDIA RTX Virtual Workstations (vWS), which provided access to the GPU. We also configured the PowerEdge server with an NVIDIA® BlueField®-2 dual-port 100 GB/s DPU. For comparison, we ran an identical PowerEdge R760 server without a GPU or DPU (see Figure 1).

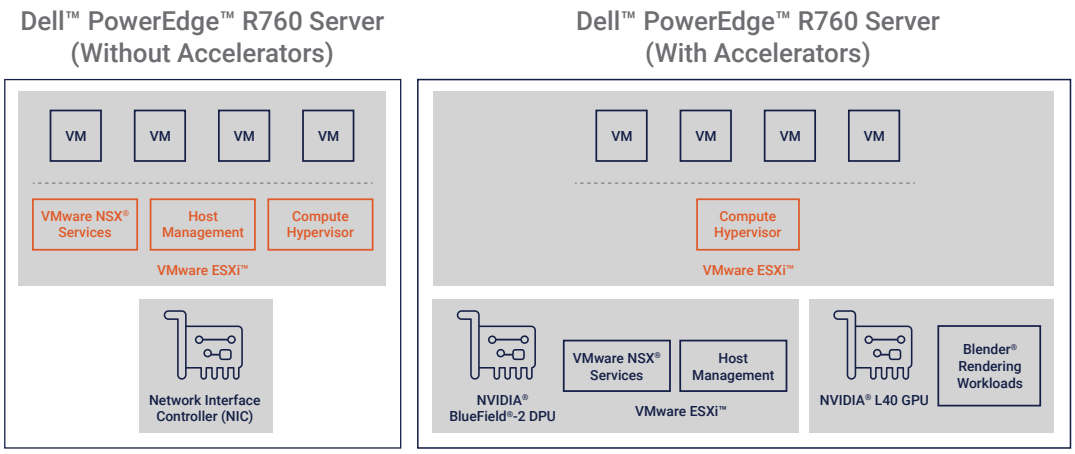

Figure 1 | Diagram showing the tested configurations, both non-accelerated (left) and accelerated (right)

For full configuration details, see Appendix B. For our step-by-step procedures, see Appendix C.

Workload Description: GPU Testing

For GPU testing, we ran Blender® video rendering workloads on vSphere virtual machines (VMs). We measured performance using the Blender benchmark score, which measures the number of path-tracing samples per minute, summed for all benchmark scenes.

We recorded the same benchmark on a single workstation and then on five workstations running concurrent rendering workloads. In both cases, we compared performance with the GPU accelerator enabled on the server to performance without a GPU enabled on the server.

Workload Description: DPU Testing

For DPU testing, we tested network performance using iPerf®, a benchmarking tool that measures the maximum achievable bandwidth on IP networks. We used iPerf to record gigabytes (GB) transferred in a 10-minute timeframe during several scenarios, each done with and without DPU acceleration:

- On one VM, while also running a concurrent Blender workload
- On one VM, without running a concurrent Blender workload
- On five VMs, without running a concurrent Blender workload

CPU Utilization

We also utilized Microsoft® PowerShell® and VMware® vSphere® PowerCLI™ to record CPU utilization on the host server while running the virtual workloads. Note that we recorded the combined CPU utilization percentages for all CPUs for each configuration, which means that the results can be greater than 100 percent. (Testing was performed on two servers with two CPUs each, for both the accelerated and non-accelerated configurations. See Appendix B for details.) This method of recording total CPU utilization helps demonstrate the clear difference between the accelerated and non-accelerated test scenarios.

Test Results

In this section, we present our test results in two sub-sections: one showing results on a single VM, the other for results collected across five VMs. (We also present the full results from all tests in table form in Appendix A.)

Test Results: Single VM

For the first test, we recorded Blender benchmark scores on a single VM, both with and without GPU acceleration. As Figure 2 shows, the accelerated configuration generated scores 36.81x (or 3,681%) higher than the non-accelerated configuration.

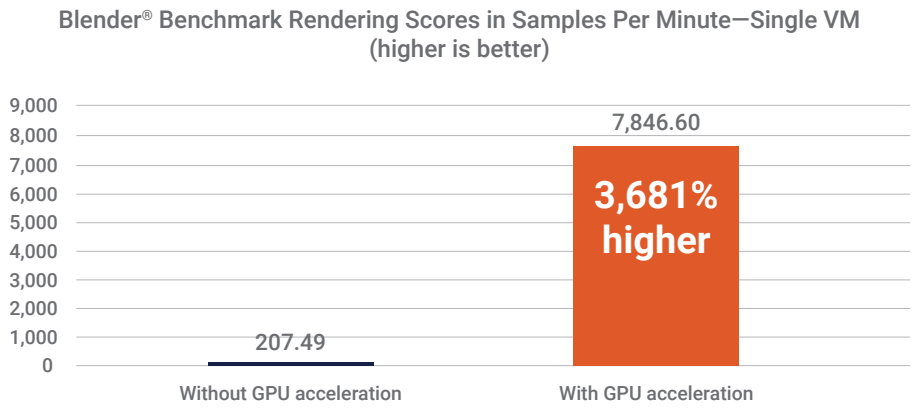

Figure 2 | Performance test results for a non-accelerated versus an accelerated Blender® benchmark workload on a single VM

We also ran iPerf in conjunction with the Blender workload to measure file-transfer performance in gigabits per second (Gbps), both with and without DPU acceleration. As Figure 3 demonstrates, the DPU accelerated configuration resulted in a 36% gain in file-transfer speed.

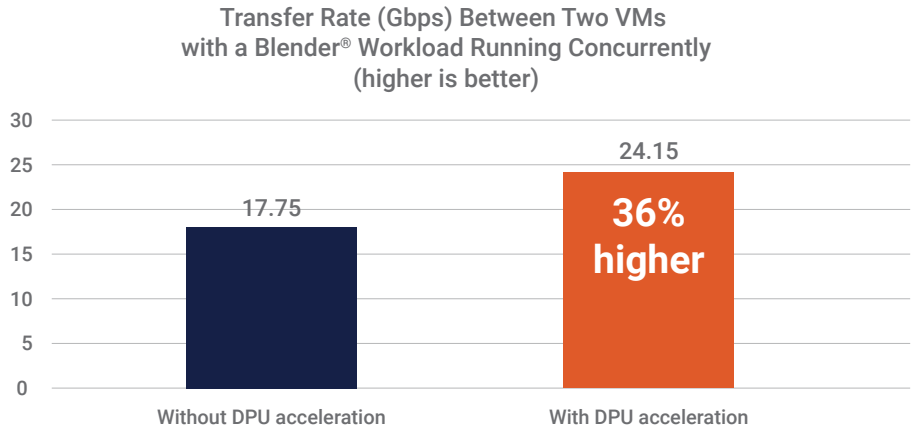

Figure 3 | File-transfer rates recorded with iPerf®, showing performance without and with DPU acceleration, measured on a single VM in conjunction with a Blender® workload

As Figure 4 shows, results were similar when we ran the iPerf test without a Blender workload, demonstrating that the DPU was primarily accelerating network traffic. (The Blender workload was primarily accelerated by the GPU, as shown in Figure 2.)

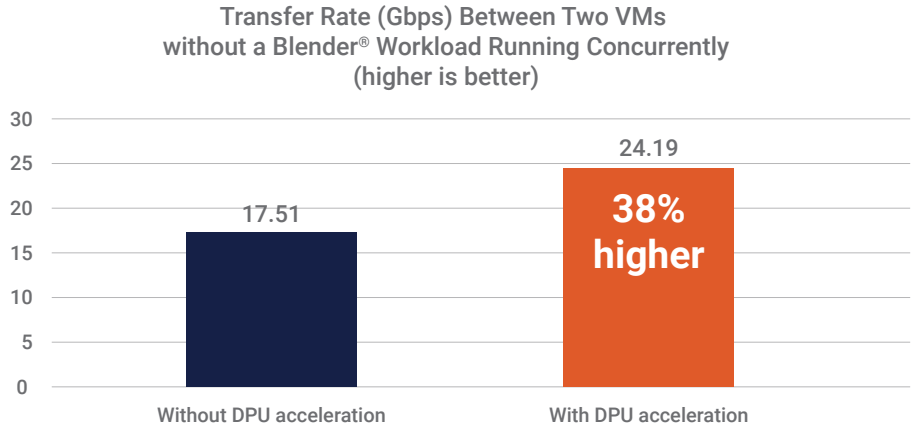

Figure 4 | File-transfer rates recorded with iPerf®, showing performance without and with DPU acceleration, measured on a single VM without a Blender® workload

In addition to significantly improving performance, the accelerators offloaded work from the CPU, as our measurements of CPU utilization show in Figure 5. The drop in CPU utilization demonstrates the ability of the DPU and GPU to free up the CPU to handle additional work for other workloads, as needed.

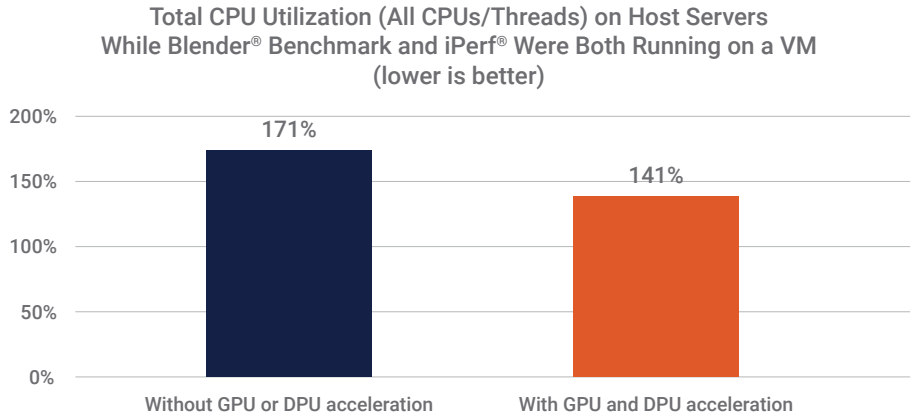

Figure 5 | Host CPU utilization (combined CPUs/threads) without and with both GPU and DPU acceleration, in conjunction with Blender® and iPerf® workloads running on a single VM

We also recorded CPU utilization rates with and without acceleration in conjunction with iPerf only (without the Blender workload). Overall CPU utilization rates were lower, as expected, without the additional workload. However, the CPU utilization rate still dropped significantly (more than 20%) when the DPU accelerator was used, as shown in Figure 6.

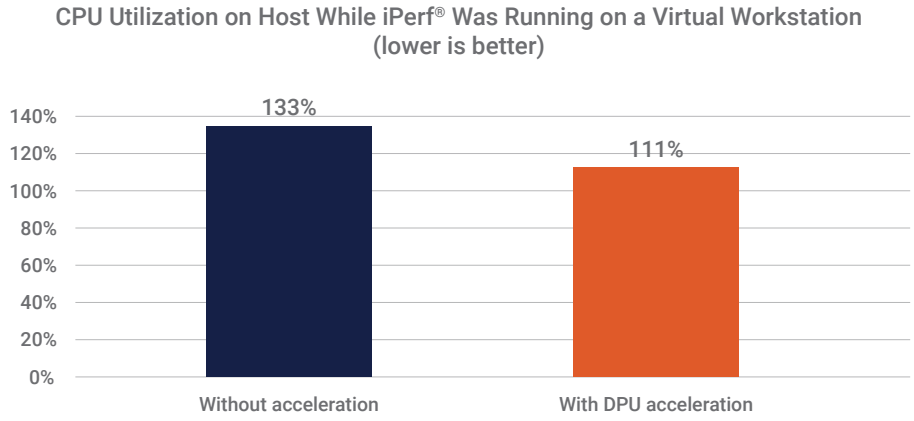

Figure 6 | Host CPU utilization (combined CPUs/threads), without and with DPU acceleration, in conjunction with iPerf® (and without Blender®)

Test Results: Five VMs

Next, we ran the Blender workload on five VMs concurrently and recorded the median score from all five benchmark results. As Figure 7 shows, this scenario resulted in scores 4.17x (or 417%) higher for the configuration using GPU acceleration, compared to the configuration without the GPU.

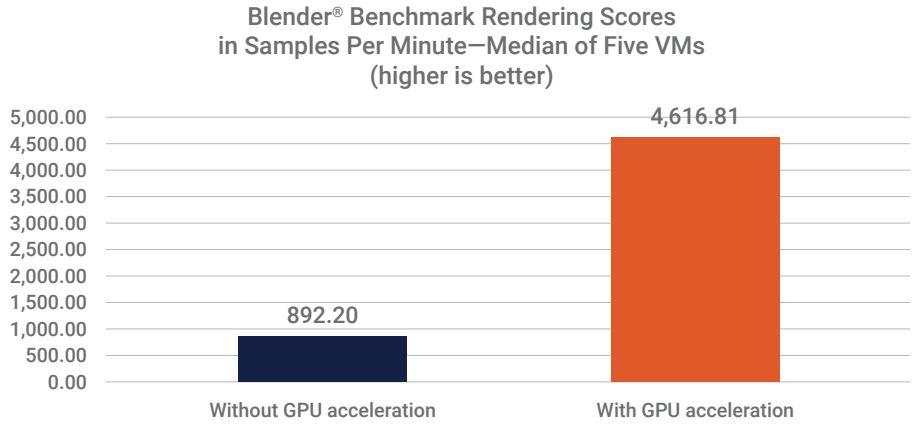

Figure 7 | Performance test results for a non-accelerated versus accelerated Blender® benchmark workload running on five VMs (median score)

CPU utilization dropped considerably when GPU and DPU accelerators were used during the five-VM Blender benchmark test, as shown in Figure 8. As with the single-VM scenario, this result shows the benefit of using accelerators to offload work from the CPU, thus freeing up the CPU for other tasks.

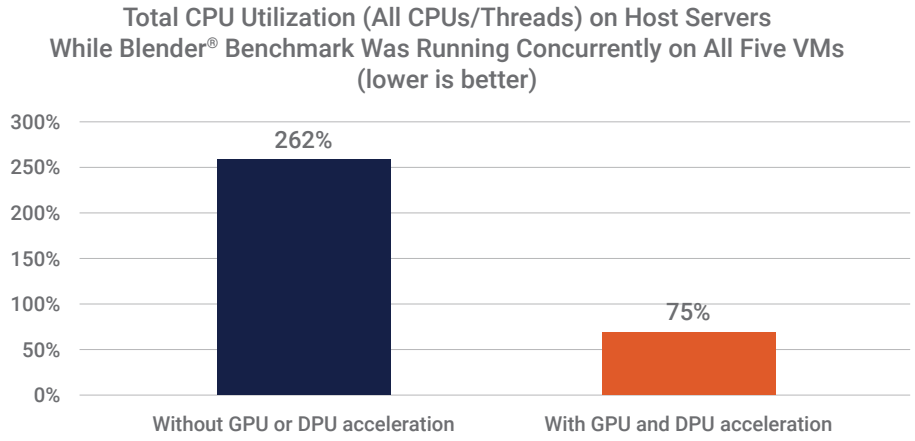

Figure 8 | Host CPU utilization (combined CPUs/threads) while the Blender® benchmark test ran on five VMs concurrently, shown both without and with accelerators

We also ran the iPerf file-transfer test on all five VMs—in this case, without the Blender workload. As with the single-VM scenario, transfer rates went up considerably (37% in this case) when DPU acceleration was used, as shown in Figure 9.

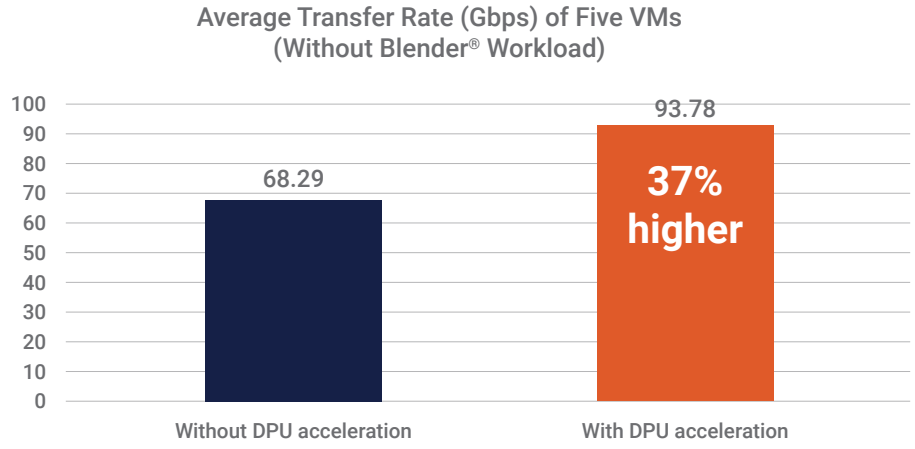

Figure 9 | Average file-transfer rates from five VMs, recorded with iPerf®, showing performance without and with DPU acceleration (without a Blender® workload)

Once again, CPU utilization dropped considerably when DPU accelerators were used during the iPerf file-transfer test, as shown in Figure 10.

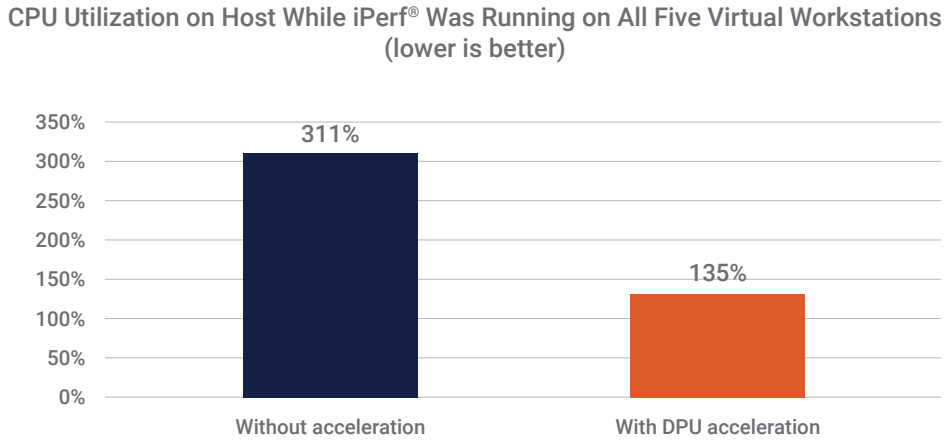

Figure 10 | Host CPU utilization (combined CPUs/threads) while iPerf® was running on all five VMs, both without and with DPU acceleration

Behind the Results

The performance gains demonstrated in our testing could be attributed to the ability of VMware vSphere 8 to support physical GPUs and DPUs.

GPU Support

By installing an NVIDIA virtual GPU (vGPU) driver in the VMware ESXi hypervisor, organizations can provide multiple VMs with shared access to the GPU. GPU access can be assigned at different levels to different VMs. In our testing, we created a shared passthrough GPU-assignment policy that was set to "Spread VMs across GPUs (best performance)."

As our testing showed, workloads that benefit from GPU acceleration can be deployed in a virtual environment while still benefiting from the physical accelerator. As a result, businesses can deploy AI/ML, 3D simulation, and other processing-intensive workloads on virtual workstations in order to ease management, reduce infrastructure requirements, simplify support, and ultimately reduce costs—all while providing enhanced performance for workers.

DPU Support

VMware vSphere 8 introduced the vSphere Distributed Services Engine, which allows offloading some of the VMware NSX® networking and security functions from the CPU to the DPU. This new feature allows distributed workloads to take advantage of resource savings, accelerated networking, and enhanced workload security, while also simplifying DPU lifecycle management with workflows integrated into vSphere.

As our iPerf testing showed, by offloading some VMware NSX networking traffic to the DPU, we were able to improve throughput and reduce CPU utilization. Organizations can make use of this offloading capability to increase workload consolidation on the host server, which can lead to a lower TCO.

Dell™ PowerEdge™ R760 Servers

The PowerEdge R760 server is an enterprise rack server designed to optimize demanding workloads like AI and ML. Powered by 4th Gen Intel® Xeon® Scalable processors, the PowerEdge R760 server is a dual-socket/2U rack server with support for up to 32 DDR5 RDIMMs at up to 4,400 megatransfers per second (MT/s) (2DPC) or up to 16 DDR5 RDIMMs at up to 4,800 MT/s (1DPC). In addition, the PowerEdge R760 server supports PCIe® Gen5 and up to 24 NVM Express® (NVMe®) drives with improved air-cooling features and optional Direct Liquid Cooling (DLC) to support increasing power and thermal requirements.

All of this makes the PowerEdge R760 server a compelling option for a wide range of workloads, including database and analytics, high-performance computing (HPC), traditional corporate IT, VDI, and AI/ML environments that require high performance, extensive storage, and GPU support.

The PowerEdge R760 server is certified for VMware vSphere® 8 and can be used to streamline deployment and management through custom VMware ESXi™ images, which are designed to simplify downloading and installing drivers, deploying patches, and accessing support.

For more information, see the PowerEdge R760 specification sheet on the Dell Technologies website.

Get Top Performance for Your Demanding Workloads

Prowess Consulting's testing demonstrated the advantages of incorporating GPUs and DPUs in a VMware vSphere 8 environment. When deployed on a supported platform, like the Dell PowerEdge R760 server we used in testing, these accelerators can provide a boost in performance for organizations that need to support compute-intensive workloads on virtual workstations.

Although our testing focused on Blender rendering and iPerf benchmark workloads, the results can be interpreted more generally to show the viability and benefits of using GPU and DPU accelerators in conjunction with vSphere deployments across any performance-hungry workload. For example, users can take advantage of GPUs to successfully and efficiently run AI, 3D modeling, database analytics, and other demanding applications in virtualized environments, which can result in faster time to results for users, along with streamlined deployments and management of workloads for IT. Moreover, by incorporating DPUs into the same vSphere deployment, organizations can accelerate VMware NSX networking and security functions while freeing up more processing capabilities on the CPU.

The benefits of a VMware software–based deployment with DPUs and GPUs go beyond performance gains. By offloading work from the CPU and reducing CPU utilization, these accelerators can help organizations increase or consolidate intensive workloads, leading to greater overall data center efficiency through a smaller server footprint, lower power and cooling costs, and—ultimately—a lower TCO.

For detailed testing methodology and the configurations used in this study, see Appendixes B and C.

Learn more about the PowerEdge servers discussed in this study:

Dell PowerEdge R760

Appendix A

Full test results from Blender benchmark and iPerf file-transfer testing, along with CPU utilization.

Test Results: Single VM

Test	Without Acceleration	With Acceleration
Blender® benchmark score (higher is better) (on one VM)	207.49	7,846.60
iPerf® file transfer (with Blender® workload running)	17.75 Gbps	24.15 Gbps
CPU utilization (with Blender® + iPerf® workloads)	171%	141%
iPerf® file transfer (without Blender® workload)	17.51 Gbps	24.19 Gbps
CPU utilization (with iPerf® only)	133%	111%

Test Results: Five VMs

Test	Without Acceleration	With Acceleration
Blender® benchmark score (higher is better) (median score across five VMs)	892.20	4,616.81
CPU utilization (with Blender® workload)	262%	75%
iPerf® file transfer (without Blender® workload)	68.29 Gbps	93.78 Gbps
CPU utilization (with iPerf® only)	311%	135%

Appendix B

The following sections provide the full configurations used for testing.

Host Servers

	Dell™ PowerEdge™ R760 Server with GPU and DPU	Dell™ PowerEdge™ R760 Server without Hardware Accelerators
Model	2 x Dell™ PowerEdge™ R760	2 x Dell™ PowerEdge™ R760
CPU	Intel® Xeon® Gold 6430 processor	Intel® Xeon® Gold 6430 processor
Number of CPUs	2	2
Cores/threads per CPU	32, 64	32, 64
Cores/threads total	64, 128	64, 128
CPU frequency	2,100 MT/s	2,100 MT/s
Storage controller 1	Dell™ Boot-Optimized Server Storage (BOSS)-N1 monolithic	Dell™ Boot-Optimized Server Storage (BOSS)-N1 monolithic
Disk	480 GB Dell™ NVM Express® (NVMe®) SK hynix® PE8010	480 GB Dell™ NVM Express® (NVMe®) SK hynix® PE8010
Number of disks	2	2
Storage controller 2	Dell™ PowerEdge RAID Controller 11 (PERC 11) H755 Front	Dell™ PowerEdge RAID Controller 11 (PERC 11) H755 Front
Disk	1,788 GB KIOXIA® PH-0N15JP-TPKK	1,788 GB KIOXIA® PH-0N15JP-TPKK
Number of disks	20	20
Storage controller 3	Not applicable (N/A)	N/A

	Dell™ PowerEdge™ R760 Server with GPU and DPU	Dell™ PowerEdge™ R760 Server without Hardware Accelerators
Disk	1.6 TB Dell™ NVMe® v2 agnostic mixed-use U.2	1.6 TB Dell™ NVMe® v2 agnostic mixed-use U.2
Number of disks	4	4
Installed memory	512 GB	512 GB
Memory DIMM	DDR5	DDR5
Memory speed	4,400 MT/s	4,400 MT/s
Number of memory DIMMs	16 of 32 slots populated	16 of 32 slots populated
DPU model	Mellanox® BlueField®-2 MT42822 integrated Mellanox® ConnectX®-6 Dx network controller	N/A
GPU model	NVIDIA® AD102GL [L40]	N/A
Network adapter 1	10 GB/25 Gb Broadcom® NetXtreme® E-Series BCM57414 remote direct memory access (RDMA) Ethernet controller	10 GB/25 Gb Broadcom® NetXtreme® E-Series BCM57414 remote direct memory access (RDMA) Ethernet controller
Operating system (OS)	VMware ESXi™ 8	VMware ESXi™ 8
OS version	8.0.2	8.0.2
OS kernel	21495797	21495797

Single-VM Configurations with GPU/DPU

	VM Configuration for Single-VM iPerf® Test	VM Configuration for Single-VM Blender® Test
VM quantity	1	1
vCPU count	16	16
Memory	64 GB	64 GB
Virtual hard disk	100 GB	200 GB
PCI device	N/A	NVIDIA® L40-24q (q notation is for the NVIDIA® vWS type)
Virtual network adapter	Distributed switch with a connection to the DPU enabled	Distributed switch with a connection to the DPU enabled
OS	Red Hat® Enterprise Linux® 8.8	Windows Server® 2022 Datacenter
OS version	Linux® 4.18.0-477.27.1.el8_8.x86_64	Build 20348.fe_release.210507-1500
Benchmark	iPerf®	Blender® CLI
Benchmark version	3.5-7.el8_8	3.1.0

Multi-VM Configurations with GPU/DPU

	VM Configuration for Multi-VM iPerf® Test	VM Configuration for Multi-VM Blender® Test
VM quantity	5	5
vCPU count	16	16
Memory	64 GB	64 GB
Virtual hard disk	100 GB	200 GB
PCI device	N/A	NVIDIA® L40-8q (q notation is for the NVIDIA® vWS type)
Virtual network adapter	Distributed switch with a connection to the DPU enabled	Distributed switch with a connection to the DPU enabled
OS	Red Hat® Enterprise Linux® 8.8	Windows Server® 2022 Datacenter
OS version	Linux 4.18.0-477.27.1.el8_8.x86_64	Build 20348.fe_release.210507-1500
Benchmark	iPerf®	Blender® CLI
Benchmark version	3.5-7.el8_8	3.1.0

Single-VM Configurations without GPU/DPU

	VM Configuration for Single VM iPerf® Test	VM Configuration for Single VM Blender® Test
VM quantity	1	1
vCPU count	16	16
Memory	64 GB	64 GB
Virtual hard disk	100 GB	200 GB
PCI device	N/A	N/A
Virtual network adapter	Distributed switch with a connection to 100 gigabit Ethernet (GbE) enabled	Distributed switch with a connection to 100 gigabit Ethernet (GbE) enabled
OS	Red Hat® Enterprise Linux® 8.8	Windows Server® 2022 Datacenter
OS version	Linux® 4.18.0-477.27.1.el8_8.x86_64	Build 20348.fe_release.210507-1500
Benchmark	iPerf®	Blender® CLI
Benchmark version	3.5-7.el8_8	3.1.0

Multi-VM Configurations without GPU/DPU

	VM Configuration for Multi-VM iPerf® Test	VM Configuration for Multi-VM Blender® Test
VM quantity	5	5
vCPU count	16	16
Memory	64 GB	64 GB
Virtual hard disk	100 GB	200 GB
PCI device	N/A	N/A
Virtual network adapter	Distributed switch with a connection to 100 GbE enabled	Distributed switch with a connection to 100 GbE enabled
OS	Red Hat® Enterprise Linux® 8.8	Windows Server® 2022 Datacenter
OS version	Linux® 4.18.0-477.27.1.el8_8.x86_64	Build 20348.fe_release.210507-1500
Benchmark	iPerf®	Blender® CLI
Benchmark version	3.5-7.el8_8	3.1.0

Appendix C

Our engineers performed testing using the following procedures:

- One Windows Server® VM performing the Blender CLI benchmark
- One Red Hat® Enterprise Linux® VM running iPerf with six threads
- One Windows Server VM performing the Blender CLI benchmark concurrently with one Red Hat Enterprise Linux VM running the iPerf benchmark with six threads
- Five Windows Server VMs performing the Blender CLI benchmark in parallel
- Five Red Hat Enterprise Linux VMs running the iPerf benchmark with six threads in parallel

During the course of our testing, we captured host metrics using VMware vSphere PowerCLI.

For the multi-VM workloads, we cloned the workload VM four times for a total of five VMs. To conduct the testing, the VMs were initially in an off state, and we turned them on at the same time. The benchmark workloads then started after five minutes of rest time. We then collected the benchmark and iPerf scores after the workload ran. This testing was conducted three times.

Dell PowerEdge R760 Server Hardware Configuration

We performed the following steps on the PowerEdge R760 servers, both for the configuration with GPU/DPU hardware accelerators and for the configuration without:

1. Launch the server into Integrated Dell™ Remote Access Controller (iDRAC).
2. Select **Install firmware updates**.
3. On the **Firmware update** page, select the Dell Technologies website, and then click **Next**.
4. On the **Firmware update: Launch firmware update** page, click **Next**.
5. If prompted for **Proxy**, click **Yes**.
6. Select the **Updates** page, review the updates, and then click **Apply**.
7. Reboot the server when directed.
8. Log in to iDRAC.
9. Select **Storage > Overview**.
    a. Select the first RAID controller.
    b. From the **Actions** drop-down menu, select **Create Virtual Disk**.
    c. On the **Set up virtual disk** page, select or enter the following:
        i. **Name: VD1 (Data)**
        ii. **Controller: PERC 11 H755 Front (Embedded)**
        iii. **Layout: RAID-5**
        iv. **Media Type: SSD**
        v. **Physical Disk Selection: New Group**
        vi. **Security: Disabled**
        vii. **Stripe Element Size: 64 KB**
        viii. **Read Policy: No Read Ahead**
        ix. **Write Policy: Write Through**
        x. **Disk Cache Policy: Disabled**
    d. Select the **Physical Disk** page, select twenty disks, and then click **Next**.
    e. On the **Virtual Disk Settings** page, click **Next**.
    f. On the **Confirmation** page, click **Add to Pending**.
    g. Click **Apply Now.**
    h. On the **Information** page, click **OK**.
10. Reboot the server.
11. From iDRAC, select **Virtual Remote Connection**.
12. Select **Virtual Media**.
13. On the **Map CD/DVD** page, select **Map device**.
14. In the **Location** field, select the **VMware ESXi 8.0.2 ISO**, and then click **Map device**.
15. Click **Close**.
16. Click **Boot**, and then select **Virtual CD/DVD/ISO**.
17. On the **VMware ESXi 8.0.2 welcome page**, press **Enter**.
18. Accept the End User License Agreement (EULA) by pressing **F11**.
19. Select **Install/Upgrade ESXi**, and then press **Enter**.
20. Navigate to the Dell™ BOSS N-1 drive via the arrow keys, press **Space** to select the drive, and then press **Enter** to continue.
21. Select a language, and then press **Enter**.
22. Enter a root password, and then press **Enter**.
23. Confirm the installation, and then press **F11** to install.
24. Remove the installation media, and then press **Enter** to reboot.
25. Upon reboot, at the **ESXi Direct User Console** window, press **F2** to customize system settings, enter root credentials, and then press **Enter**.
26. Navigate to **Configure Management Network**, and then **IPv4 Configuration**.
27. Select **Set static IPv4 address and network configuration** via the **Space** key, and then press **Enter** to confirm.
28. From a client device, enter the previously assigned static IP address, and then log in to the VMware ESXi host with the root credentials.

29. On the left pane, navigate to **Manage > System > Date & Time**:
    a. Click **Edit NTP Settings**.
    b. Select **Use Network Time Protocol (enable NTP client)**.
    c. Set **NTP service startup policy to Start and stop with host**.
    d. Set the IP or FQDN of the NTP server(s).
    e. Click **Save**.
    f. At the top of the page, navigate to **Services**.
    g. Ensure the Network Time Protocol daemon (ntpd) service is in a **Running** state.
30. On the left pane, navigate to **Storage**:
    a. Select **New datastore**.
    b. On the **Select creation type** window, select **Create new VMFS datastore**, and then click **Next**.
    c. On the **Select device page**, select the **RAID5** device, and then name the new datastore.

NVIDIA® L40 GPU Installation and Configuration

Use the following steps to install and configure the services to support the NVIDIA L40 GPU on the GPU/DPU VMware vSphere hosts. For the purposes of this testing, our engineers utilized a trial account to download the NVIDIA Enterprise licenses.

1. From the NVIDIA Enterprise License portal, download the **Complete vGPU 15.4 package for VMware vSphere 8.0 including supported guest drivers**.
2. In VMware vCenter®, navigate to the left pane, select **Storage**, and then select the applicable datastore.
3. At the top of the page, click **Datastore Browser**, and then upload the vGPU driver.
4. In the **VMware ESXi Host** graphical user interface (GUI), right-click **Host**, and then select **Services > Enable SSH**.
5. Create a Secure Shell (SSH) connection into the host with root credentials.
6. Enable maintenance mode with the following commands, and then install **mgmt.-daemon.vib**:

```
esxcli system maintenanceMode set --enable true
esxcli software vib install -d /vmfs/volumes/<datastore location>/<folder>/<file>.zip
```

7. Reboot the server and repeat the process with the NVIDIA GRID™ vGPU driver.
8. Disable maintenance mode with the following command:

```
esxcli system maintenanceMode set --enable false
```

9. Reboot the host.
10. In VMware vCenter, navigate to the host, and then, at the top of the page, select **Configure > Graphics**.
11. Under **Graphics Devices**, select **NVIDIA L40/NVIDIA L40 CNX**, and then click **Edit**.
12. Select **Shared Direct**, and then click **OK**.
13. Under **Host Graphics**, on the right side of the page, select **Edit**.
14. On the **Edit Host Graphics Settings** page, select **Shared**.
15. For **Shared passthrough GPU assignment policy**, select **Spread VMs across GPUs (best performance)**.
16. Click **OK** to confirm the changes.

NVIDIA® BlueField®-2 DPU Installation and Configuration

Use the following steps to install and configure the services to support the NVIDIA BlueField-2 DPU on the GPU/DPU VMware vSphere hosts. For the purposes of this testing, our engineers were provided with trial access to the VMware NSX Enterprise Plus product.

1. From VMware vCenter, right-click the **Client Server** at the host level, and then select **Deploy OVF Template**.
2. In the **Select an OVF template** page, select **Local file**, and then click **Upload files**.
3. Select the **NSX-Embedded-Unified-Appliance** OVA file.
4. Click **Next**.
5. On the **Select a name and folder** page, enter a name for the VMware NSX appliance.
6. Select a folder to deploy the appliance to.
7. Click **Next**.
8. On the **Select a compute resource** page, select the client server.
9. Click **Next**.
10. On the **Review details** page, click **Next**.
11. On the **Configuration** page, select **Medium**.
12. Click **Next**.

13. On the **Select storage** page, select **datastore2**.
14. On the **Select networks** page, select **VM Network**.
15. Click **Next**.
16. On the **Customize template** page, fill in the following required parameters:
    a. Create a password.
    b. Confirm the password.
    c. Create a hostname for the VM.
    d. Enter an IPv4 address for the first interface.
    e. Enter the netmask for the first interface.
    f. Enter DNS server(s).
    g. Enter NTP server(s).
17. On the **Ready to complete** page, click **Finish**.
18. Power on the VMware NSX appliance.

Distributed Virtual Switch Creation and Configuration

Use the following steps to create and configure the distributed virtual switch to support the NVIDIA BlueField-2 DPU on the GPU/DPU VMware vSphere hosts:

1. In vCenter, at the data center level, right-click and select **Distributed Switch > New Distributed Switch**.
2. On the **Name and location** page, enter a name for the new distributed switch.
3. Click **Next**.
4. On the **Select version** page, select **8.0.0 - ESXi 8.0 and later**.
5. Click **Next**.
6. On the **Configure settings** page, enter the following parameters:
    a. Set **Network Offloads compatibility** to **NVIDIA BlueField**.
    b. Set **Number of uplinks** to **2**.
    c. Network input/output (I/O) is disabled when network offloads compatibility is enabled.
    d. Enter a name for the new distributed port group.
7. On the **Ready to complete** page, click **Finish**.
8. Right-click the distributed switch, and then select **Add and Manage Hosts**.
9. On the **Select task** page, select **Add hosts**.
10. Click **Next**.
11. On the **Select hosts** page, select the NVIDIA BlueField platform–compatible hosts.
12. Click **Next**.
13. On the **Manage physical adapters** page, assign uplink(s) on the applicable vmnic.
14. Click **Next**.
15. Skip **Manage VM kernel adapters**, and then click **Next**.
16. Skip **Migrate VM networking**, and then click **Next**.
17. On the **Ready to complete** page, click **Finish**.
18. In a web browser, navigate to the IP address of the VMware NSX appliance.
19. From the home page, navigate to **System > Fabrics > Profiles > Uplink Profiles**.
20. Select **Add Profile**.
21. Enter a name for the new profile (for example, **smnic-uplink-profile**).
22. Under **Teamings**, add **Active/Passive Uplinks**.
23. Enter a desired VLAN value.
24. Set **MTU** to **1700**.
25. Navigate to **System > Fabric > Nodes > Transport Node Profiles**.
26. Click **Add Transport Node Profile**.
27. Enter a name for the new transport node profile.
28. Click **Set**.
29. Click **Add Host Switch**.
30. Select **vCenter Appliance**.
31. Select the previously created virtual distributed switch.

32. Select the previously created uplink profile.
33. Select **IP Address Type**.
34. Set **IPv4 Assignment**.
35. Expand **Advanced Configuration**.
36. In the **Mode** drop-down menu, select **Enhanced Datapath – Standard**.
37. Navigate to **System > Fabric > Hosts > Other Nodes**.
38. Select the DPU-enabled hosts.
39. Click **Configure NSX**.
40. Click **Next**.
41. Click **Add Host Switch**.
42. In the **VDS** drop-down menu, select the distributed virtual switch created previously.
43. Under **Transport Zone**, ensure the previously-created transport zone is selected.
44. In the **Uplink Profile** drop-down menu, select the previously-created uplink profile.
45. In the **IP Address Type** field, select **IPv4**.
46. In the **IPv4 assignment** drop-down menu, select **Static**.
47. Assign an **IPv4 Address**, **Subnet Mask**, and **IPv4 Gateway.**
48. Expand the **Advanced Configuration** menu.
49. Select **Enhance Datapath - Standard**.
50. Under **Teaming Policy Uplink Mapping**, select **Uplink priority**.
51. Click **Add**.
52. Click **Finish**.
53. Monitor the installation via the **NSX Configuration** column.

Virtual Switch Creation

Perform the following steps to create and configure a virtual switch to be used with the 100 GbE network adapters on the hosts without a DPU or GPU:

1. In vCenter, right-click at the host level and select **Add Networking**.
2. Select the **Physical Network Adapter** radio button, and then click **Next**.
3. Select the **New standard switch** radio button.
4. In the **MTU (Bytes)** field, enter **9000** for the value, and then click **Next**.
5. Select the applicable unclaimed 100 GbE adapter(s), and then click **Next**.
6. On the **Ready to complete** page, review the selections, and then select **Finish**.

NVIDIA DLS Appliance Deployment and Configuration

Use the following steps to create and configure the NVIDIA delegated license server (DLS) to support the NVIDIA L40 GPU on the VMware vSphere hosts with a GPU/DPU:

1. In the vCenter client, right-click the infrastructure host or a DPU/GPU-enabled host, and then select **Deploy OVF template**.
2. Upload and specify the DLS OVA.
3. Specify the name for the VM, and then select the folder where you would like to place the VM.
4. Specify the compute resource on which you want to deploy the appliance.
5. On the **Review details** pane, review the configuration.
6. Select the datastore on which you want to deploy the appliance.
7. On the **Select networks** pane, specify the port group.
8. On the **Ready to complete** pane, confirm the configuration, and then click **OK**.
9. From the vCenter client, open the remote or web console of the appliance.
10. Log in using the account dls_system; there is no need to provide a password.
11. From the command-line interface (CLI), execute the following script:

```
/etc/adminscripts/set-static-ip-cli.sh
```

12. Follow the script steps to complete the DLS networking configuration.
13. Browse to the IP address of the appliance, and then select **First time setup**.
14. Specify the password for the dls_admin user account.
15. Sign in to the NVIDIA application hub and navigate to the **NVIDIA Licensing Portal**.

16. From the menu on the left, select **License Servers**.
17. Click the green **Create server** button.
18. Enter a name for the license server.
19. Select the **Virtual Workstation** licensing entitlement, and then click **Create server**.
20. In the administrative console of the DLS appliance, select **Download DLS Instance Token**.
21. In the NVIDIA application hub, from the left menu, select **Service Instances**.
22. On the right, select **Actions**, and then click **Upload DLS instance token**.
23. Select the **New Installation** radio button, and then upload the DLS instance token from the previous step.
24. In the **Service Instances** window, find the pending service instance, and then, in the **Actions** menu, select **Register**.

Windows Server® VM Creation

Perform the following steps to create a VM to support testing with the Blender benchmark workload on Windows Server 2022:

1. In vCenter, navigate to the left pane, select **Storage**, and then select the applicable datastore.
2. At the top of the page, click **Datastore Browser**.
3. Create a new folder, and then assign it a name.
4. Click the newly created folder, and then click **Upload**.
5. Upload the Windows Server 2022 Datacenter ISO.
6. On the left pane, navigate to **Virtual Machines**.
7. On the select creation type, select **Create a new virtual machine**, and then click **Next**.
8. Select a name and guest OS:
    a. Enter a name for the VM.
    b. In the **Compatibility** drop-down, select **ESXi 8.0 U2 virtual machine**.
    c. In the **Guest OS family** drop-down, select **Windows**.
    d. In the **Guest OS Version**, select **Microsoft Windows Server 2022(64-bit)**.
    e. Click **Next**.
9. Select **Storage**:
    a. Select an applicable datastore.
    b. Click **Next**.
10. Customize the following settings:
    a. **CPU: 16**
    b. **Memory: 64GB**
    c. **Hard disk 1: 200GB**
11. Assign **Network adapter 1** to **VM Network**, and leave **Adapter type** as **VMXNET 3**.
12. For the DPU-/GPU-enabled workload:
    a. Select **Add New Device > Network Adapter**.
    b. In the **New Network Adapter** drop-down menu, select the VMware NSX vNIC port group.
13. For the non-DPU-/GPU-enabled workload:
    a. Select **Add New Device > Network Adapter**.
    b. In the **New Network Adapter** drop-down menu, select the 100 GbE vNIC port group.
14. From the **CD/DVD Drive** drop-down menu, select **Datastore ISO file**:
    a. Expand additional settings for **CD/DVD Drive**, and then click **Browse**.
    b. Navigate to **datastore 1**, and then select the uploaded **Windows Server 2022** ISO.
15. For the DPU-/GPU-enabled workload only, perform the following steps:
    a. Select **Add New Device > PCI Device**:
        i. For the single-VM workload, select **nvidia_l40-8q**.
        ii. For the multi-VM workload, select **nvidia_l40-24q**.
16. Click **Next** and review the VM configuration.
17. Click **Finish**.
18. Confirm VM creation under recent tasks.
19. Power on the VM.

Windows Server 2022 Datacenter Installation and Configuration

Use the following steps to install and configure Windows Server 2022 Datacenter:

1. On the initial setup screen, click **Next**.
2. Click **Install Now**.
3. On the **Operating system version selection** page, select **Windows Server 2022 Datacenter Evaluation (Desktop Experience)**.
4. Click **Next**.
5. Accept the **Applicable notices and license terms**, and then click **Next**.
6. Select **Custom installation**.
7. Select an applicable drive on which to install the OS, and then click **Next**.
8. Once installation is complete, the system will restart and prompt for creation of an administrator password.
9. Log in to the administrator profile.
10. Open **Server Manager**:
    a. Select **Remote Desktop**, and then enable **Remote Desktop**.
11. Open a web browser:
    a. Download the Blender benchmark from https://opendata.blender.org.
    b. Extract the contents to a preferred location.
    c. Open Notepad, and then enter the following:

```
@ECHO OFF
set SAVESTAMP=%DATE:/=-%@%TIME::=-%
set SAVESTAMP=%SAVESTAMP: =%
set SAVESTAMP=%SAVESTAMP:,=.%.txt
timeout 300
cd <preferred extraction location>
timeout 2
.\benchmark-launcher-cli.exe --blender-version 4.0.0 benchmark monster junkshop classroom --device-
type OPTIX --json >> c:\temp\blender_%SAVESTAMP%.txt
```

    d. Select **File > Save As**, and then enter ***<FileName>*.bat**.
    e. For **File type**, select **All File Types**.
    f. Select **Save**.
12. Use the following guide to enable automatic logon: https://learn.microsoft.com/en-us/troubleshoot/windows-server/user-profiles-and-logon/turn-on-automatic-logon.
13. From the **Start** menu, search for and open **Task Scheduler**.
14. Right-click **Task Scheduler (Local)**:
    a. Click **Create Task**.
    b. Under **General**, name the task.
    c. In the **Configure for** drop-down menu, select **Windows Server 2022**.
    d. Under **Triggers**, click **New**.
    e. Select the **One-time** radio button.
    f. Clear all of the **Advanced settings** checkboxes except **Enabled**.
    g. Click **OK**.
    h. Under **Actions**, click **New**.
    i. In the **New Action** window, in the drop-down menu, select **Start a program.**
    j. Click **Browse**.
    k. Navigate to the previously created .bat file.
    l. Click **OK**.
    m. Under **Conditions**, clear the **Start the task only if the computer is on AC power** checkbox.
    n. Click **OK** to save the settings and create the scheduled task.

Red Hat® Enterprise Linux® 8.8 VM Creation

Perform the following steps to create a VM to support testing with the iPerf workload on Red Hat Enterprise Linux 8.8:

1. From VMware vCenter, navigate to **Storage**, and then select applicable datastore.
2. At the top of the page, click **Datastore Browser**.

3. Create a new folder, and then assign it a name.
4. Click the newly created folder, and then click **Upload**.
5. Upload the Red Hat Enterprise Linux 8.4 ISO.
6. On the left pane, navigate to **Virtual Machines**.
7. On the select creation type, select **Create a new virtual machine**, and then click **Next**.
8. Select a name and guest OS:
    a. Enter a name for the VM.
    b. In the **Compatibility** drop-down menu, select **ESXi 8.0 U2 virtual machine**.
    c. In the **Guest OS family** drop-down menu, select **Linux**.
    d. In the **Guest OS Version** drop-down menu, select **Red Hat Enterprise Linux 8 (64-bit)**.
    e. Click **Next**.
9. Select **Storage**:
    a. Select an applicable datastore.
    b. Click **Next**.
10. Customize the following settings:
    a. **CPU: 16**
    b. **Memory: 64GB**
    c. **Hard disk 1: 200GB**
11. Assign **Network adapter 1** to **VM Network**, and leave **Adapter type** as **VMXNET 3**.
12. For the DPU-/GPU-enabled workload:
    a. Select **Add New Device > Network Adapter**.
    b. In the **New Network Adapter** drop-down menu, select the VMware NSX vNIC port group.
13. For the non-DPU-/GPU-enabled workload:
    a. Select **Add New Device > Network Adapter**.
    b. In the **New Network Adapte**r drop-down menu, select the 100 GbE vNIC port group.
14. From the **CD/DVD Drive** drop-down menu, select **Datastore ISO file**:
    a. Expand additional settings for **CD/DVD Drive**, and then click **Browse**.
    b. Navigate to **datastore 1**, and then select the uploaded **Red Hat Enterprise Linux 8.4** ISO.
15. Click **Next**, and then review the VM configuration.
16. Click **Finish**.
17. Confirm VM creation under recent tasks.
18. Power on the VM.

Red Hat Enterprise Linux 8.8 OS Installation and Configuration

Use the following steps to install and configure Red Hat Enterprise Linux 8.8:

1. On the **Welcome** page, select a language, and then click **Continue**.
2. Select **Software Selection** options, and then, under **Additional Software for Selected Environment**, select **Development Tools**.
3. Under **User Settings**, select **Root Password**, and then enter a root password.
4. Re-enter the root password to confirm, and then click **Done**.
5. Select **Installation Destination**.
6. At the bottom of the page, under **Storage Configuration**, select **Custom**.
7. Click **Done**.
8. Under **Manual Partitioning**, click the **Click here to create them automatically** link.
9. Allocate the following partition storage values:
    a. **/home: 18 GiB**
    b. **/root: 64 GiB**
    c. **/boot/efi: 600 MiB**
    d. **/boot: 1024 MiB**
    e. **Swap: 16 GiB**
    f. Click **Done.**
10. Confirm the partition reconfiguration.

11. Select **Begin Installation**.
12. After installation is complete, reboot the VM.
13. Once rebooted, on the **Initial Setup** page, select **License Information**, and then click **I Accept The License Agreement**.
14. Click **Done**.
15. Click **Finish Configuration**.
16. On the **Welcome** page, click **Next**.
17. On the **Privacy** page, click **Next**.
18. On the **Online Accounts** page, click **Skip**.
19. On the **About You** page, enter a **User Name (Admin)**, and then click **Next**.
20. Set an account password, and then confirm the password.
21. Click **Next**.
22. Select **Start Using Red Hat Enterprise Linux**.
23. Open a **Terminal**, change user to **root**, and then run the following command to register the system:

```
subscription-manager register
```

24. Fill in the **Username** field, and then press **Enter**.
25. Enter a **Password**, and then press **Enter**.
26. The prompt will confirm that the system has been successfully registered.
27. Run the following command to update packages:

```
dnf update -y
```

28. Run the following command to stop and disable the firewall service:

```
systemctl stop firewalld && systemctl disable firewalld
```

29. Disable SELINUX with the following command:

```
nano /etc/selinux/config
```

30. In the text editor, modify SELINUX:

```
SELINUX=disabled
```

31. Save the file and reboot the guest OS.
32. At the top right of the desktop, select the **Power/Network settings** icon.
33. For the DPU-/GPU-enabled workload, perform the following steps:
    a. Select **VMware NSX vNIC**, and then select **Wired Settings**.
    b. Select the gear icon to edit the network configuration with the following details:
        i. **IPv4 Method: Manual**
        ii. **Address: 192.168.2.X**
        iii. **Netmask: 255.255.255.0**
        iv. **Gateway**: Leave blank
    c. Select **Use this connection only for resources on its network**.
34. For the non-DPU-/GPU-enabled workload, perform the following steps:
    a. Select the **100 GbE vNIC**, and then select **Wired Settings**.
    b. Select the gear icon to edit the network configuration with the following details:
        i. **IPv4 Method: Manual**
        ii. **Address: 192.168.2.X**
        iii. **Netmask: 255.255.255.0**
        iv. **Gateway:** Leave blank
    c. Select **Use this connection only for resources on its network**.
35. Download iPerf 3.1.3 from https://iperf.fr/.
36. Run the following command to install iPerf:

```
dnf install <iPerf_Filename>
```

iPerf Test Process

Perform the following steps to initiate the iPerf testing process:

1. On the iPerf server, run the following command to initiate listening on the specified ports:

```
iperf3 -s -p 5201 & iperf3 -s -p 5202 & iperf3 -s -p 5203 & iperf3 -s -p 5204 & iperf3 -s -p 5205 &
iperf3 -s -p 5206 &
```

2. On the iPerf client server, run the following command to initiate testing on the previously specified ports for the given duration:

```
iperf3 -c <server IP> -t 600 -p 5201 --logfile <log file location.log> & iperf3 -c <server IP> -t 600 -p 5202 --logfile <log file location.log> & iperf3 -c <server IP> -t 600 -p 5203 --logfile <log file location.log> & iperf3 -c <server IP> -t 600 -p 5204 --logfile <log file location.log> & iperf3 -c <server IP> -t 600 -p 5205 --logfile <log file location.log> & iperf3 -c <server IP> -t 600 -p 5206 --logfile <log file location.log> &
```

3. Wait until the benchmark is complete, and then collect the benchmark results.

Blender® Test Process

Perform the following steps to initiate the Blender test process:

1. Shut down the Windows Server VM.
2. Wait five minutes.
3. From VMware vCenter, select the Windows Server VM(s), right-click, and then select **Power On**. The benchmark will run automatically.
4. Wait until the benchmark is complete, and then collect the benchmark results.

[1] Based on testing by Prowess Consulting, completed January 2024. See Appendix A for full test results.

[2] The Next Platform. "Economics and the Inevitability of the DPU." November 2022.

[3] VMware NSX. "DPU-based Acceleration for NSX: Overview." August 2022.

0224/230085