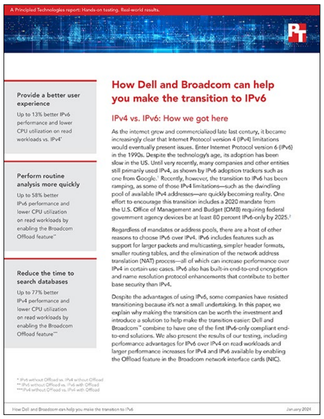
How Dell and Broadcom can help you make the transition to IPv6

IPv4 vs. IPv6: How we got here

The science behind the report:

How Dell and Broadcom can help you make the transition to IPv6

This document describes what we tested, how we tested, and what we found. To learn how these facts translate into real-world benefits, read the report How Dell and Broadcom can help you make the transition to IPv6.

We concluded our hands-on testing on November 15, 2023. During testing, we determined the appropriate hardware and software configurations and applied updates as they became available. The results in this report reflect configurations that we finalized on November 14, 2023 or earlier. Unavoidably, these configurations may not represent the latest versions available when this report appears.

Our results

To learn more about how we have calculated the wins in this report, go to http://facts.pt/calculating-and-highlighting-wins. Unless we state otherwise, we have followed the rules and principles we outline in that document.

Table 1: IPv6 vs. IPv4 performance on a write workload with Offload off. Higher IOPS and MB/sec and lower CPU utilization are better. Source: Principled Technologies.

Write workload, Offload off				
IP version	**Block size**	**IOPS**	**MB/sec**	**Percentage CPU utilization**
IPv4	256K	8,696.1	2,174.01	4.9
IPv6	256K	8,752.1	2,188.02	4.9
IPv6 % improvement		**0.64%**	**0.64%**	**0.00%**
IPv4	64K	34,862.7	2,178.92	6.6
IPv6	64K	34,972.1	2,185.76	6.5
IPv6 % improvement		**0.31%**	**0.31%**	**1.51%**

Table 2: IPv6 vs. IPv4 performance on a read workload with Offload off. Higher IOPS and MB/sec and lower CPU utilization are better. Source: Principled Technologies.

Read workload, Offload off				
IP version	**Block size**	**IOPS**	**MB/sec**	**Percentage CPU utilization**
IPv4	256K	19,987.8	4,996.95	14.1
IPv6	256K	22,752.4	5,688.09	13.6
IPv6 % improvement		**13.83%**	**13.83%**	**3.54%**
IPv4	64K	73,194.1	4,574.63	13.9
IPv6	64K	80,392.4	5,024.53	12.5
IPv6 % improvement		**9.83%**	**9.83%**	**10.07%**

Table 3: IPv6 performance on a write workload with Offload off and Offload on. Higher IOPS and MB/sec and lower CPU utilization are better. Source: Principled Technologies.

Write workload, IPv6				
	Block size	**IOPS**	**MB/sec**	**Percentage CPU utilization**
Offload off	256K	8,752.1	2,188.02	4.9
Offload on	256K	8,615.5	2,153.88	2
Offload on % improvement		**-1.56%**	**-1.56%**	**59.18%**
Offload off	64K	34,972.1	2,185.76	6.5
Offload on	64K	34,895.6	2,180.97	3.4
Offload on % improvement		**-0.21%**	**-0.21%**	**47.69%**

Table 4: IPv6 performance on a read workload with Offload off and Offload on. Higher IOPS and MB/sec and lower CPU utilization are better. Source: Principled Technologies.

Read workload IPv6				
	Block size	**IOPS**	**MB/sec**	**Percentage CPU utilization**
Offload off	256K	22,752.4	5,688.09	13.6%
Offload on	256K	35,983.4	8,995.86	8.3%
Offload on % improvement		**58.15%**	**58.15%**	**38.97%**
Offload off	64K	80,392.4	5,024.53	12.5%
Offload on	64K	100,840.9	6,302.55	7.3%
Offload on % improvement		**25.43%**	**25.43%**	**41.60%**

Table 5: IPv4 performance on a write workload with Offload off and Offload on. Higher IOPS and MB/sec and lower CPU utilization are better. Source: Principled Technologies.

Write workload IPv4				
	Block size	**IOPS**	**MB/sec**	**Percentage CPU utilization**
Offload off	256K	8,696.1	2,174.01	4.9
Offload on	256K	8,596.0	2,148.99	2.0
Offload on % improvement		**-1.56%**	**-1.15%**	**59.18%**
Offload off	64K	34,862.7	2,178.92	6.6
Offload on	64K	34,727.8	2,170.49	3.6
Offload on % improvement		**-0.21%**	**-0.38%**	**45.45%**

Table 6: IPv4 performance on a read workload with Offload off and Offload on. Higher IOPS and MB/sec and lower CPU utilization are better. Source: Principled Technologies.

Read workload IPv4				
	Block size	**IOPS**	**MB/sec**	**Percentage CPU utilization**
Offload off	256K	19,987.8	4,996.95	14.1
Offload on	256K	35,559.3	8,889.81	8.1
Offload on % improvement		**58.15%**	**77.90%**	**42.55%**
Offload off	64K	73,194.1	4,574.63	13.9
Offload on	64K	101,545.6	6,346.60	7.4
Offload on % improvement		**25.43%**	**38.73%**	**46.76%**

System configuration information

Table 7: Detailed information on the systems we tested.

System configuration information	Dell PowerEdge R660
BIOS name and version	Dell 1.4.4
Non-default BIOS settings	None
Operating system name and version/build number	SUSE Linux Enterprise Server 15 SP4 kernel 5.14.21-150400.22-default
Date of last OS updates/patches applied	11/16/2023
Power management policy	Performance Per Watt (OS)
Processor	
Number of processors	2
Vendor and model	Intel® Xeon® Gold 5418Y
Core count (per processor)	24
Core frequency (GHz)	2.00
Stepping	8
Memory module(s)	
Total memory in system (GB)	256
Number of memory modules	8
Vendor and model	Hynix® HMCG88AEBRA115N
Size (GB)	32
Type	DDR5 2Rx8
Speed (MHz)	4,800
Speed running in the server (MHz)	4,400
Storage controller (A)	
Vendor and model	Dell PERC H755 Front
Cache size (GB)	8
Firmware version	52.21.0-4606
Local storage (type A)	
Number of drives	2
Drive vendor and model	Micron® MTFDDAK480TDS
Drive size (GB)	480
Drive information (speed, interface, type)	6Gbps SATA SSD
Network adapter	
Vendor and model	Broadcom® BCM57508
Number and type of ports	2x100G QSFP
Driver version	22.31.13.70

System configuration information	Dell PowerEdge R660
Cooling fans	
Vendor and model	Gold
Number of cooling fans	6
Power supplies	
Vendor and model	Dell 1100W
Number of power supplies	2
Wattage of each (W)	1,100

Table 8: Detailed information on the systems we tested.

System configuration information	Dell PowerStore 1200T
Software version	3.5.0.1 (Release, Build 2083289, 2023-06-30 00:24:17, Retail)
Number of storage shelves	1
Number of data drives	12x NVMe SSD 7x NVMe SCM 2x NVMe NVRAM
Drive part number	005052921 005053027 005053499
Drive size (GB)	3,840 375 8

How we tested

To determine the performance differences between IPv4 and IPv6 in a "real-world" environment, we used two Dell PowerEdge R660 servers configured with SLES15 SP4 to send data sets of diverse sizes to a Dell PowerStore 1200T storage array via VDBench for Linux using standard Linux tools, NVMe/TCP, and/or NFS as the transport protocols. The data traversed multiple switches that we configured to provide NAT and packet fragmentation within a heterogeneous multi-hop network.

Each of the two Dell PowerEdge R660 servers had 256 GB of memory and eight 10GB PowerStore volumes, as well as two 100GbE ports via Broadcom 57508 NICs. We configured one server to use IPv4 and the other to use IPv6. The storage array we used was a PowerStore 1200T providing 16x 100GB volumes for the two PowerEdge servers, with 2x 100GbE ports to interface with the servers. The network consisted of 2x Dell S5248F Ethernet/IP NAT switches and 1x Dell S5232F Ethernet/IP Core switch. We configured each host interface to access a dedicated storage interface to create a 1:1 non-oversubscribed configuration.

We used VDBench for Linux as the IO-generating tool, running 100% Sequential Read and 100% Sequential Write workloads at block sizes of 64 KB and 256 KB. We completed these tests with the networking MTU set to 9,000 bytes on the SUT (system under test) for both the IPv4-only and IPv6-only servers. We ran all these set of tests both with TCP Offload Enabled and with TCP Offload Disabled.

Configuring the test environment

Configuring SUSE Linux Enterprise 15 Service Pack 4 for the SUT

1. On the 100Gb NICs, assign a static IP address, and set the MTU to 9,000.
2. Install VDBench and Java on the SLES15SP4 servers.
3. Connect to PowerStore via NVMe™/TCP using the following example commands:

   - IPv4

```
nvme connect -t tcp -n [powerstore volume?] -a [IPv4 address] -s 4420 -l -1
nvme connect -t tcp -n [powerstore volume?] -a [IPv4 address 2] -s 4420 -l -1
```

   - IPv6

```
nvme connect -t tcp -n [powerstore volume?] -a [IPv6 address] -s 4420 -l -1
nvme connect -t tcp -n [powerstore volume?] -a [IPv6 address 2] -s 4420 -l -1
```

4. Run the command `nvme list-subsys`
5. The output should look like this example.

   - IPv4

```
nvme-subsys0 - NQN=nqn.1988-11.com.dell:powerstore:00:c12e476b8792C4276E48
\
 +- nvme0 tcp traddr=192.168.10.10,trsvcid=4420 live non-optimized
 +- nvme1 tcp traddr=192.168.20.20,trsvcid=4420 live optimized
 +- nvme0 tcp traddr=192.168.10.10,trsvcid=4420 live optimized
 +- nvme1 tcp traddr=192.168.20.20,trsvcid=4420 live non-optimized
 +- nvme0 tcp traddr=192.168.10.10,trsvcid=4420 live non-optimized
 +- nvme1 tcp traddr=192.168.20.20,trsvcid=4420 live optimized
 +- nvme0 tcp traddr=192.168.10.10,trsvcid=4420 live non-optimized
 +- nvme1 tcp traddr=192.168.20.20,trsvcid=4420 live optimized
 +- nvme0 tcp traddr=192.168.10.10,trsvcid=4420 live optimized
 +- nvme1 tcp traddr=192.168.20.20,trsvcid=4420 live non-optimized
 +- nvme0 tcp traddr=192.168.10.10,trsvcid=4420 live optimized
 +- nvme1 tcp traddr=192.168.20.20,trsvcid=4420 live non-optimized
 +- nvme0 tcp traddr=192.168.10.10,trsvcid=4420 live non-optimized
 +- nvme1 tcp traddr=192.168.20.20,trsvcid=4420 live optimized
 +- nvme0 tcp traddr=192.168.10.10,trsvcid=4420 live optimized
 +- nvme1 tcp traddr=192.168.20.20,trsvcid=4420 live non-optimized
```

- IPv6

```
nvme-subsys0 - NQN=nqn.1988-11.com.dell:powerstore:00:c12e476b8792C4276E48
\
 +- nvme0 tcp traddr=2003:beef:a:10::10,trsvcid=4420 live non-optimized
 +- nvme1 tcp traddr=2003:beef:a:20::20,trsvcid=4420 live optimized
 +- nvme0 tcp traddr=2003:beef:a:10::10,trsvcid=4420 live optimized
 +- nvme1 tcp traddr=2003:beef:a:20::20,trsvcid=4420 live non-optimized
 +- nvme0 tcp traddr=2003:beef:a:10::10,trsvcid=4420 live non-optimized
 +- nvme1 tcp traddr=2003:beef:a:20::20,trsvcid=4420 live optimized
 +- nvme0 tcp traddr=2003:beef:a:10::10,trsvcid=4420 live optimized
 +- nvme1 tcp traddr=2003:beef:a:20::20,trsvcid=4420 live non-optimized
 +- nvme0 tcp traddr=2003:beef:a:10::10,trsvcid=4420 live optimized
 +- nvme1 tcp traddr=2003:beef:a:20::20,trsvcid=4420 live non-optimized
 +- nvme0 tcp traddr=2003:beef:a:10::10,trsvcid=4420 live non-optimized
 +- nvme1 tcp traddr=2003:beef:a:20::20,trsvcid=4420 live optimized
 +- nvme0 tcp traddr=2003:beef:a:10::10,trsvcid=4420 live non-optimized
 +- nvme1 tcp traddr=2003:beef:a:20::20,trsvcid=4420 live optimized
 +- nvme0 tcp traddr=2003:beef:a:10::10,trsvcid=4420 live optimized
 +- nvme1 tcp traddr=2003:beef:a:20::20,trsvcid=4420 live non-optimized
```

Preparing scripts for VDBench

1. Prepare VDBench and bash script files to automate testing for 100% Sequential Read and 100% Sequential Write on block sizes of 64 KB and 256 KB (see the Scipts section).

Running the tests

Running the VDBench tests

1. Run VDBench with 100% Sequential Read and 100% Sequential Write on block sizes of 64 K and 256 K using the ./run_vdbench.sh script. Run first with TCP Offload enabled, and run again with TCP Offload disabled.

```
./run_vdbench.sh <WORKLOAD> <BLOCKSIZE> <OFFLOAD>
<WORKLOAD>: seqread | seqwrite
<BLOCKSIZE>: 64K | 256K
<OFFLOAD>: on | off
```

2. To enable TCP Offload on NICs p1p1 and p1p2, use the following commands:

```
ethtool -K p1p1 tcp-segmentation-offload on generic-segmentation-offload on generic-receive-offload on
rx-vlan-offload on tx-vlan-offload on hw-tc-offload on rx-udp_tunnel-port-offload on
ethtool -K p1p2 tcp-segmentation-offload on generic-segmentation-offload on generic-receive-offload on
rx-vlan-offload on tx-vlan-offload on hw-tc-offload on rx-udp_tunnel-port-offload on
```

3. To disable TCP Offload on NICs p1p1 and p1p2, use the following commands:

```
ethtool -K p1p1 tcp-segmentation-offload off generic-segmentation-offload off generic-receive-offload
off rx-vlan-offload off tx-vlan-offload off hw-tc-offload off rx-udp_tunnel-port-offload off
ethtool -K p1p2 tcp-segmentation-offload off generic-segmentation-offload off generic-receive-offload
off rx-vlan-offload off tx-vlan-offload off hw-tc-offload off rx-udp_tunnel-port-offload off
```

4. Perform these tests three times, noting the median throughput run.

Scripts

run_vdbench.sh

```
#!/bin/bash
WORKLOAD=${1:-seqread}
BLOCKSIZE=${2:-64K}
OFFLOAD=${3:-off}
TIMESTAMP=$(date '+%Y%m%d%H%M%S')
NICS="p1p1 p1p2"
INTERVAL=5
RUNTIME=300
SAMPLES=$((RUNTIME/INTERVAL))
TEMPLATE="${WORKLOAD}-${BLOCKSIZE}.dat"
if ! [ -f ${TEMPLATE} ]; then
  echo "Template file missing: ${TEMPLATE}"
  exit
fi
OUTPUT_DIR=output/$(hostname -s)_offload-${OFFLOAD}_${WORKLOAD}-${BLOCKSIZE}_${TIMESTAMP}/
mkdir -p ${OUTPUT_DIR}
for NIC in ${NICS}; do
  ethtool -K ${NIC} tcp-segmentation-offload ${OFFLOAD} generic-segmentation-offload ${OFFLOAD} generic-
receive-offload ${OFFLOAD} rx-vlan-offload ${OFFLOAD} tx-vlan-offload ${OFFLOAD} hw-tc-offload ${OFFLOAD}
rx-udp_tunnel-port-offload ${OFFLOAD}
  ethtool -k ${NIC} > ${OUTPUT_DIR}/ethtool-k_${NIC}.txt
done
ip addr > ${OUTPUT_DIR}/ip_addr.txt
nvme list > ${OUTPUT_DIR}/nvme_list.txt
nvme list-subsys > ${OUTPUT_DIR}/nvme_list-subsys.txt
dmidecode > ${OUTPUT_DIR}/dmidecode.txt
sync
sleep ${INTERVAL}
./vdbench -f ${TEMPLATE} -o ${OUTPUT_DIR} &
sleep ${INTERVAL}
sar -o ${OUTPUT_DIR}/sar.bin ${INTERVAL} $((SAMPLES-2)) 2>&1 > ${OUTPUT_DIR}/sar.txt &
wait
```

seqread-64k.dat

```
sd=A-1,lun=/dev/nvme0n8,openflags=o_direct
sd=A-2,lun=/dev/nvme0n7,openflags=o_direct
sd=A-3,lun=/dev/nvme0n6,openflags=o_direct
sd=A-4,lun=/dev/nvme0n5,openflags=o_direct
sd=A-5,lun=/dev/nvme0n4,openflags=o_direct
sd=A-6,lun=/dev/nvme0n3,openflags=o_direct
sd=A-7,lun=/dev/nvme0n2,openflags=o_direct
sd=A-8,lun=/dev/nvme0n1,openflags=o_direct

wd=wd1,sd=A-*,seekpct=seq,rdpct=100,xfersize=64K
rd=rd1,wd=wd*,interval=5,iorate=999999,forthread=32,elapsed=300
```

seqread-256k.dat

```
sd=A-1,lun=/dev/nvme0n8,openflags=o_direct
sd=A-2,lun=/dev/nvme0n7,openflags=o_direct
sd=A-3,lun=/dev/nvme0n6,openflags=o_direct
sd=A-4,lun=/dev/nvme0n5,openflags=o_direct
sd=A-5,lun=/dev/nvme0n4,openflags=o_direct
sd=A-6,lun=/dev/nvme0n3,openflags=o_direct
sd=A-7,lun=/dev/nvme0n2,openflags=o_direct
sd=A-8,lun=/dev/nvme0n1,openflags=o_direct

wd=wd1,sd=A-*,seekpct=seq,rdpct=100,xfersize=256K
rd=rd1,wd=wd*,interval=5,iorate=999999,forthread=32,elapsed=300
```

seqwrite-64k.dat

```
sd=A-1,lun=/dev/nvme0n8,openflags=o_direct
sd=A-2,lun=/dev/nvme0n7,openflags=o_direct
sd=A-3,lun=/dev/nvme0n6,openflags=o_direct
sd=A-4,lun=/dev/nvme0n5,openflags=o_direct
sd=A-5,lun=/dev/nvme0n4,openflags=o_direct
sd=A-6,lun=/dev/nvme0n3,openflags=o_direct
sd=A-7,lun=/dev/nvme0n2,openflags=o_direct
sd=A-8,lun=/dev/nvme0n1,openflags=o_direct

wd=wd1,sd=A-*,seekpct=seq,rdpct=0,xfersize=64K
rd=rd1,wd=wd*,interval=5,iorate=999999,forthread=32,elapsed=300
```

seqwrite-256k.dat

```
sd=A-1,lun=/dev/nvme0n8,openflags=o_direct
sd=A-2,lun=/dev/nvme0n7,openflags=o_direct
sd=A-3,lun=/dev/nvme0n6,openflags=o_direct
sd=A-4,lun=/dev/nvme0n5,openflags=o_direct
sd=A-5,lun=/dev/nvme0n4,openflags=o_direct
sd=A-6,lun=/dev/nvme0n3,openflags=o_direct
sd=A-7,lun=/dev/nvme0n2,openflags=o_direct
sd=A-8,lun=/dev/nvme0n1,openflags=o_direct

wd=wd1,sd=A-*,seekpct=seq,rdpct=0,xfersize=256K
rd=rd1,wd=wd*,interval=5,iorate=999999,forthread=32,elapsed=300
```

Read the report at https://facts.pt/2ml5Pbm

This project was commissioned by Dell Technologies.

Principled Technologies®

Facts matter.®

Principled Technologies is a registered trademark of Principled Technologies, Inc.
All other product names are the trademarks of their respective owners.

DISCLAIMER OF WARRANTIES; LIMITATION OF LIABILITY:
Principled Technologies, Inc. has made reasonable efforts to ensure the accuracy and validity of its testing, however, Principled Technologies, Inc. specifically disclaims any warranty, expressed or implied, relating to the test results and analysis, their accuracy, completeness or quality, including any implied warranty of fitness for any particular purpose. All persons or entities relying on the results of any testing do so at their own risk, and agree that Principled Technologies, Inc., its employees and its subcontractors shall have no liability whatsoever from any claim of loss or damage on account of any alleged error or defect in any testing procedure or result.

In no event shall Principled Technologies, Inc. be liable for indirect, special, incidental, or consequential damages in connection with its testing, even if advised of the possibility of such damages. In no event shall Principled Technologies, Inc.'s liability, including for direct damages, exceed the amounts paid in connection with Principled Technologies, Inc.'s testing. Customer's sole and exclusive remedies are as set forth herein.