The science behind the report:

Unlock more mixed storage performance on Dell PowerEdge R750 servers with Broadcom PCIe Gen4 RAID controllers

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 Unlock more mixed storage performance on Dell PowerEdge R750 servers with Broadcom PCIe Gen4 RAID controllers.

We concluded our hands-on testing on April 27, 2022. 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 March 28, 2022 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 https://facts.pt/calculating-and-highlighting-wins. Unless we state otherwise, we have followed the rules and principles we outline in that document.

Table 1: Results of our 4KB random read workload FIO benchmark testing.

Random read testing results	Dell PowerEdge™ R750	Dell PowerEdge™ R740xd
Mixed SAS/NVMe™ drive configurations		
2 SAS and 2 NVMe drives Thousands of IOPS (kIOPS)	3,305	1,893
4 SAS and 4 NVMe drives (kIOPS)	6,597	3,790
6 SAS and 6 NVMe drives (kIOPS)	9,869	5,300
16 SAS and 8 NVMe drives (kIOPS)	13,230	5,533

Table 2: Results of our 4KB random write workload FIO benchmark testing.

Random write testing results	Dell PowerEdge R750	Dell PowerEdge R740xd
Mixed SAS/NVMe drive configurations		
2 SAS and 2 NVMe drives (kIOPS)	2,054	1,827
4 SAS and 4 NVMe drives (kIOPS)	4,098	3,653
6 SAS and 6 NVMe drives (kIOPS)	6,113	5,432
16 SAS and 8 NVMe drives (kIOPS)	9,068	7,074

Table 3: Results of our 1MB sequential read workload FIO benchmark testing.

Sequential read testing results	Dell PowerEdge R750	Dell PowerEdge R740xd
Mixed SAS/NVMe drive configurations		
2 SAS and 2 NVMe drives Throughput (GiB/s)	14.1	8.2
4 SAS and 4 NVMe drives Throughput (GiB/s)	28.2	16.5
6 SAS and 6 NVMe drives Throughput (GiB/s)	42.2	24.6
16 SAS and 8 NVMe drives Throughput (GiB/s)	62.4	30.1

Table 4: Results of our 1MB sequential write workload FIO benchmark testing.

Sequential write testing results	Dell PowerEdge R750	Dell PowerEdge R740xd
Mixed SAS/NVMe drive configurations		
2 SAS and 2 NVMe drives Throughput (GiB/s)	9.2	7.8
4 SAS and 4 NVMe drives Throughput (GiB/s)	18.5	15.7
6 SAS and 6 NVMe drives Throughput (GiB/s)	27.8	23.6
16 SAS and 8 NVMe drives Throughput (GiB/s)	42.5	31.6

System configuration information

Table 5: Detailed information on the systems we tested.

Server configuration information	Dell PowerEdge R750	Dell PowerEdge R740xd
BIOS name and version	Dell 1.5.5	Dell 2.13.3
Operating system name and version/ build number	Red Hat® Enterprise Linux® release 8.5 (Ootpa)/4.18.0-348.20.1.el8_5.x86_64	Red Hat® Enterprise Linux® release 8.5 (Ootpa)/4.18.0-348.20.1.el8_5.x86_64
Date of last OS updates/patches applied	4/18/22	4/18/22
Processor		
Number of processors	2	2
Vendor and model	Intel® Xeon® Gold 6330	Intel® Xeon® Gold 6252L
Core count (per processor)	28	24
Core frequency (GHz)	2.00	2.10
Stepping	6	7
Memory module(s)		
Total memory in system (GB)	256	192
Number of memory modules	16	12
Vendor and model	Samsung® M393A2K43DB3-CWE	Micron 18ASF2G72PDZ-3G2R1
Size (GB)	16	16
Type	DDR4	DDR4
Speed (MHz)	3,200	3,200
Speed running in the server (MHz)	2,933	2,933
Storage controller		
Vendor and model	PERC H755 front SAS (PCIe® Gen4)	PERC H740P Adapter (PCIe® Gen3)
Firmware version	52.16.1-4158	51.16.0-4076
SAS drives		
Number of drives	16	16
Drive vendor and model	KIOXIA KRM6VVUG3T84	KIOXIA KRM6VVUG3T84
Drive size (GB)	3,840	3,840
Drive information	12 Gbps SAS SSD	12 Gbps SAS SSD
Local storage		
Number of drives	8	8
Drive vendor and model	Dell Ent NVMe CM6 MU 3.2 TB	Dell Ent NVMe P5600 MU U.2 3.2TB
Drive size (GB)	3,200	3,200
Drive information	NVMe SSD	NVMe SSD

Server configuration information	Dell PowerEdge R750	Dell PowerEdge R740xd
Network adapter		
Vendor and model	Broadcom NetXtreme BCM5720 2-port Gigabit Ethernet PCIe, Broadcom BCM57414 NetXtreme-E 10Gb/25Gb RDMA Ethernet Controller	Broadcom NetXtreme BCM5720 2-port Gigabit Ethernet PCIe, Broadcom BCM57414 NetXtreme-E 10Gb/25Gb RDMA Ethernet Controller
Number and type of ports	2 x 1GbE, 2 x 10GbE	2 x 1GbE, 2 x 10GbE
Cooling fans		
Vendor and model	Dell 7002W-A00	Dell XN44V-A00
Number of cooling fans	6	6
Power supplies		
Vendor and model	Dell 0CYHHJ	Dell 04V8KD
Number of power supplies	2	2
Wattage of each (W)	1,400	700

How we tested

Both systems were dual-processor. Each processor connected to 4 NVMe drives, and the SAS RAID controller connected to the first processor. We tested four mixed SAS/NVMe drive configurations, with 4, 8, 12, and 24 drives total in each configuration (2x SAS + 2x NVMe, 4x SAS + 4x NVMe, 6x SAS + 6x NVMe, 16x SAS + 8x NVMe). The R740xd and R750 had different NVMe drives.

Installing Red Hat 8.5

Install the operating system on both the Dell PowerEdge R740xd and Dell PowerEdge R750 systems.

1. Open a browser window, and connect to the iDRAC.
2. Log into the iDRAC.
3. To open it, click Virtual Console.
4. Click Connect Virtual Media.
5. Next to Map CD/DVD, click Browse.
6. Browse to the ISO for Red Hat 8.5.
7. Click Map Device.
8. Click Boot.
9. Click the Virtual CD/DVD/ISO text.
10. Click yes.
11. Click Power, and boot the machine.
12. At the Red Hat Enterprise Linux boot menu, press Up, select Install Red Hat Enterprise Linux 8.5.0, and press Enter.
13. At the Welcome screen, make sure English is selected, and click Continue.
14. At the Installation Summary screen, click Time & Date.
15. To adjust your location, use the down arrows next to Region and City, and click Done.
16. At the Installation Summary screen, click Software Selection.
17. At the Software Selection screen, click Minimal Install, and click Done.
18. At the Installation Summary screen, click Installation Destination.
19. At the Installation Destination screen, select the internal JBOSS drive. Leave Automatic selected for the Storage Configuration, and click Done.
20. At the Installation Summary screen, click Network & Host Name.
21. At the Network & Host Name screen, where it says Host Name, enter `R740xd` or `R750`, and click Apply.
22. At the top-right of the screen, next to Ethernet information, click the OFF slider to ON, allow the NIC connect, pick up an IP address from DHCP, and click Done.
23. Click Begin Installation.
24. At the Configuration screen, click Root Password. Next to Root Password and Confirm, type in your preferred password, and click Done.
25. When the Reboot button appears, click it.

Preparing scripts and running FIO tests

Prepare FIO and bash script files to automate testing and gathering results on both systems. We created 24-drive configuration files and varied the iodepth, numjobs, and other parameters to find the configuration that yielded the maximum results. We modified the parameters for each workload type (e.g., random read) to get the best performance for that workload. We experimented with different parameters for each drive configuration but achieved the best results using the same parameters with each drive configuration.

1. Create the following bash scripts, and install the appropriate software:

    a. runtest.sh:

```
#!/bin/bash

test_config=$1
testno=${2:-0}
stat_collection_time=${3:-30}
hostname=`hostname -s`
cur_time=$(date +"%Y.%m.%d_%H.%M")
host=`echo $hostname | cut -d "." -f1`
OUTDIR=output_${host}/STATS_${cur_time}_${host}_${test_config}_${testno}
mkdir -p $OUTDIR
echo "Directory name is $OUTDIR"
echo "Starting nmon"
```

```
nmon -F ./$OUTDIR/${host}_${test_config}_${testno}.nmon -s5 -J
echo "Starting fio"
fio --output-format=normal --terse-version=3 ./${test_config}.cfg --output=./$OUTDIR/${test_config}_
TEST${testno}.txt &
fiopid=$!
echo "Waiting 60 seconds for system to be in a steady state"
sleep 60
echo "Getting system stats for $stat_collection_time seconds"
mpstat -P ALL 1 > ./$OUTDIR/mpstat_out.txt &
pidmpstat=$!
dstat -t -v --nocolor > ./$OUTDIR/dstat_out.txt &
piddstat=$!
iostat -xmt 1 > ./$OUTDIR/iostat_out.txt &
pidiostat=$!
sleep $stat_collection_time
echo "Killing mpstat"
#ipid=`ps aux | grep mpstat | grep -v grep | awk '{print $2}'`
kill -9 $pidmpstat
echo "Killing dstat"
#pid=`ps aux | grep dstat | grep -v grep | awk '{print $2}'`
kill -9 $piddstat
echo "Killing iostat"
#pid=`ps aux | grep iostat | grep -v grep | awk '{print $2}'`
kill -9 $pidiostat
echo "Waiting for fio to complete"
while kill -0 $fiopid 2> /dev/null; do
     sleep 10
done
#mv mpstat_out.txt ${OUTDIR}/mpstat_out.txt
#mv dstat_out.txt ${OUTDIR}/dstat_out.txt
#mv iostat_out.txt ${OUTDIR}/iostat_out.txt
wait
pkill nmon
cat ./$OUTDIR/${test_config}_TEST${testno}.txt
echo "Test $OUTDIR_complete"
```

b. condition_mixed16+8.sh

```
#!/bin/bash
CONFIG=mixed16+8
sync
# SAS discard
for i in {a..p}; do blkdiscard -v /dev/sd${i} & done ; wait ; sync
# NVMe discard
for i in {0..7}; do blkdiscard -v /dev/nvme${i}n1 & done ; wait ; sync
# Fill
sleep 1
fio fill_${CONFIG}.cfg
```

c. Get_Sys_Info_NVMeDir.sh:

```
#!/bin/sh'
################################################################################'
### THE FOLLOWING VARIABLES CAN BE MODIFIED FOR YOUR SPECIFIC TEST REQUIREMENTS ###'
################################################################################'
'
'
storcli="storcli64"'
controller_number=0'
hostname='hostname''
cur_time=$(date "+%Y.%m.%d-%H.%m")'
host='echo $hostname | cut -d "." -f1''
OUTDIR=SysInfo_${RANDOM}_${host}_${cur_time}'
echo "*** All results and system information can be found in the $OUTDIR directory."'
mkdir -p $OUTDIR'
```

```
echo'
'
############################################################'
############## Collecting System Information #############'
############################################################'
'
collect_system_info() {'
echo'
echo "########################################"'
echo "####     COLLECTING SYSTEM INFO       ####"'
echo "########################################"'
echo'
'
'
'
# Collect System Info'
echo -e 'hostname' >> $OUTDIR/Sys_Config.txt'
echo -e "OS: 'cat /etc/redhat-release'\n" >> $OUTDIR/Sys_Config.txt'
echo -e "Proc Version: 'cat /proc/version'\n" >> $OUTDIR/Sys_Config.txt'
echo -e "\nMemory Information:" >> $OUTDIR/Sys_Config.txt'
cat /proc/meminfo >> $OUTDIR/Sys_Config.txt'
echo -e "\nProcessor Information:" >> $OUTDIR/Sys_Config.txt'
cat /proc/cpuinfo >> $OUTDIR/Sys_Config.txt'
echo -e "\nPCI Information:" >> $OUTDIR/Sys_Config.txt'
lspci -vvv  >> $OUTDIR/PCI_Config.txt'
echo -e "\nSystem DMI Information:" >> $OUTDIR/Sys_Config.txt'
dmidecode >> $OUTDIR/Sys_Config.txt'
echo -e "\nChecking SCSI MQ Settings, use_blk_mq is set to: 'cat /sys/module/scsi_mod/parameters/
use_blk_mq'\n" >> $OUTDIR/Sys_Config.txt'
numactl --hardware >> $OUTDIR/NUMA.txt  2> /dev/null'
nvme list >> $OUTDIR/nvmelist.txt'
'
rm -rf storcli.log'
echo "*** System specific information can be found in $OUTDIR\\SystemInfo directory"'
echo'
sleep 1'
}'
'
'
'
############################################################'
###################### MAIN PROGRAM ######################'
############################################################'
'
'
'
collect_system_info'
'
Enter the following commands'
yum -y install sysstat'
yum -y install pcp-system-tools'
yum -y install numactl'
chmod a+x *.sh
```

2. Create the FIO configuration files with *.cfg as the file extension. We provide an example file used for each workload here, along with listing what parameters we adjusted for individual tests. All files we provide are for the 24-drive tests.

   a. 4k_random_read_mixed16+8.cfg

```
[global]
include global.cfg
include randread.cfg

[sas0]
filename=/dev/sda

[sas1]
```

```
filename=/dev/sdb

[sas2]
filename=/dev/sdc

[sas3]
filename=/dev/sdd

[sas4]
filename=/dev/sde

[sas5]
filename=/dev/sdf

[sas6]
filename=/dev/sdg

[sas7]
filename=/dev/sdh

[sas8]
filename=/dev/sdi

[sas9]
filename=/dev/sdj

[sas10]
filename=/dev/sdk

[sas11]
filename=/dev/sdl

[sas12]
filename=/dev/sdm

[sas13]
filename=/dev/sdn

[sas14]
filename=/dev/sdo

[sas15]
filename=/dev/sdp

[nvme0]
filename=/dev/nvme0n1
numa_cpu_nodes=0

[nvme1]
filename=/dev/nvme1n1
numa_cpu_nodes=0

[nvme2]
filename=/dev/nvme2n1
numa_cpu_nodes=0

[nvme3]
filename=/dev/nvme3n1
numa_cpu_nodes=0

[nvme4]
filename=/dev/nvme4n1
numa_cpu_nodes=1

[nvme5]
filename=/dev/nvme5n1
numa_cpu_nodes=1
```

```
[nvme6]
filename=/dev/nvme6n1
numa_cpu_nodes=1

[nvme7]
filename=/dev/nvme7n1
numa_cpu_nodes=1
```

b. 4k_random_write_mixed16+8.cfg

```
[global]
include global.cfg
include randwrite.cfg

[sas0]
filename=/dev/sda

[sas1]
filename=/dev/sdb

[sas2]
filename=/dev/sdc

[sas3]
filename=/dev/sdd

[sas4]
filename=/dev/sde

[sas5]
filename=/dev/sdf

[sas6]
filename=/dev/sdg

[sas7]
filename=/dev/sdh

[sas8]
filename=/dev/sdi

[sas9]
filename=/dev/sdj

[sas10]
filename=/dev/sdk

[sas11]
filename=/dev/sdl

[sas12]
filename=/dev/sdm

[sas13]
filename=/dev/sdn

[sas14]
filename=/dev/sdo

[sas15]
filename=/dev/sdp

[nvme0]
filename=/dev/nvme0n1
numa_cpu_nodes=0
```

```
[nvme1]
filename=/dev/nvme1n1
numa_cpu_nodes=0

[nvme2]
filename=/dev/nvme2n1
numa_cpu_nodes=0

[nvme3]
filename=/dev/nvme3n1
numa_cpu_nodes=0

[nvme4]
filename=/dev/nvme4n1
numa_cpu_nodes=1

[nvme5]
filename=/dev/nvme5n1
numa_cpu_nodes=1

[nvme6]
filename=/dev/nvme6n1
numa_cpu_nodes=1

[nvme7]
filename=/dev/nvme7n1
numa_cpu_nodes=1
```

c. 1m_sequential_read_mixed16+8.cfg

```
[global]
include global.cfg
include seqread.cfg

[sas0]
filename=/dev/sda

[sas1]
filename=/dev/sdb

[sas2]
filename=/dev/sdc

[sas3]
filename=/dev/sdd

[sas4]
filename=/dev/sde

[sas5]
filename=/dev/sdf

[sas6]
filename=/dev/sdg

[sas7]
filename=/dev/sdh

[sas8]
filename=/dev/sdi

[sas9]
```

```
filename=/dev/sdj

[sas10]
filename=/dev/sdk

[sas11]
filename=/dev/sdl

[sas12]
filename=/dev/sdm

[sas13]
filename=/dev/sdn

[sas14]
filename=/dev/sdo

[sas15]
filename=/dev/sdp

[nvme0]
filename=/dev/nvme0n1
numa_cpu_nodes=0

[nvme1]
filename=/dev/nvme1n1
numa_cpu_nodes=0

[nvme2]
filename=/dev/nvme2n1
numa_cpu_nodes=0

[nvme3]
filename=/dev/nvme3n1
numa_cpu_nodes=0

[nvme4]
filename=/dev/nvme4n1
numa_cpu_nodes=1

[nvme5]
filename=/dev/nvme5n1
numa_cpu_nodes=1

[nvme6]
filename=/dev/nvme6n1
numa_cpu_nodes=1

[nvme7]
filename=/dev/nvme7n1
numa_cpu_nodes=1
```

d. 1m_sequential_write_mixed16+8.cfg

```
[global]
include global.cfg
include seqwrite.cfg

[sas0]
filename=/dev/sda

[sas1]
filename=/dev/sdb

[sas2]
filename=/dev/sdc
```

```
[sas3]
filename=/dev/sdd

[sas4]
filename=/dev/sde

[sas5]
filename=/dev/sdf

[sas6]
filename=/dev/sdg

[sas7]
filename=/dev/sdh

[sas8]
filename=/dev/sdi

[sas9]
filename=/dev/sdj

[sas10]
filename=/dev/sdk

[sas11]
filename=/dev/sdl

[sas12]
filename=/dev/sdm

[sas13]
filename=/dev/sdn

[sas14]
filename=/dev/sdo

[sas15]
filename=/dev/sdp

[nvme0]
filename=/dev/nvme0n1
numa_cpu_nodes=0

[nvme1]
filename=/dev/nvme1n1
numa_cpu_nodes=0

[nvme2]
filename=/dev/nvme2n1
numa_cpu_nodes=0

[nvme3]
filename=/dev/nvme3n1
numa_cpu_nodes=0

[nvme4]
filename=/dev/nvme4n1
numa_cpu_nodes=1

[nvme5]
filename=/dev/nvme5n1
numa_cpu_nodes=1

[nvme6]
filename=/dev/nvme6n1
numa_cpu_nodes=1
```

```
[nvme7]
filename=/dev/nvme7n1
numa_cpu_nodes=1
```

e. fill_mixed16+8.cfg

```
[global]
name=1m_seq_write
#scramble_buffers=1
refill_buffers=1
buffered=0
sync=0
ioengine=libaio
norandommap
do_verify=0
randrepeat=0
random_generator=tausworthe64
offset=0
thinktime=0
time_based=0
direct=1
overwrite=0
group_reporting=1
bs=1m
blockalign=1m
rw=write
#numa_cpu_nodes=0
cpus_allowed_policy=shared
iodepth=256
numjobs=1

[sda]
filename=/dev/sda

[sdb]
filename=/dev/sdb

[sdc]
filename=/dev/sdc

[sdd]
filename=/dev/sdd

[sde]
filename=/dev/sde

[sdf]
filename=/dev/sdf

[sdg]
filename=/dev/sdg

[sdh]
filename=/dev/sdh

[sdi]
filename=/dev/sdi

[sdj]
filename=/dev/sdj

[sdk]
filename=/dev/sdk

[sdl]
```

```
filename=/dev/sdl

[sdm]
filename=/dev/sdm

[sdn]
filename=/dev/sdn

[sdo]
filename=/dev/sdo

[sdp]
filename=/dev/sdp

[nvme0]
filename=/dev/nvme0n1
numa_cpu_nodes=0

[nvme1]
filename=/dev/nvme1n1
numa_cpu_nodes=0

[nvme2]
filename=/dev/nvme2n1
numa_cpu_nodes=0

[nvme3]
filename=/dev/nvme3n1
numa_cpu_nodes=0

[nvme4]
filename=/dev/nvme4n1
numa_cpu_nodes=1

[nvme5]
filename=/dev/nvme5n1
numa_cpu_nodes=1

[nvme6]
filename=/dev/nvme6n1
numa_cpu_nodes=1

[nvme7]
filename=/dev/nvme7n1
numa_cpu_nodes=1
```

f. global.cfg

```
scramble_buffers=1
buffered=0
sync=0
ioengine=libaio
norandommap
do_verify=0
randrepeat=0
random_generator=tausworthe64
offset=0
thinktime=0
time_based=1
direct=1
overwrite=0
group_reporting=1
#numa_cpu_nodes=0
cpus_allowed_policy=shared
ramp_time=120s
runtime=60s
```

g. randread.cfg

```
name=4k_random_read
bs=4k
blockalign=4k
rw=randread
iodepth=32
numjobs=8
```

h. randwrite.cfg

```
name=4k_random_write
bs=4k
blockalign=4k
rw=randwrite
iodepth=32
numjobs=8
```

i. seqread.cfg

```
name=1m_sequential_read
bs=1m
blockalign=1m
rw=read
iodepth=64
numjobs=4
```

j. seqwrite.cfg

```
name=1m_sequential_write
bs=1m
blockalign=1m
rw=write
iodepth=32
numjobs=4
```

3. After the system boots, run your tests using the bash scripts by entering the following commands at the console:

```
./condition_mixed<drive-config>.sh
./Get_Sys_Info_NVMeDir.sh
./runtest.sh <desired fio test config> <test number>
```

4. View your results in the created, appropriately named directory.

Read the report at https://facts.pt/4Xw0EX6 ▶

This project was commissioned by Dell Technologies.