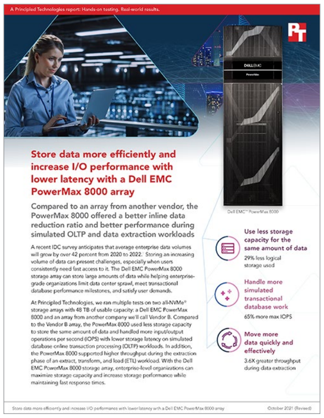

The science behind the report:

Store data more efficiently and increase I/O performance with lower latency with a Dell EMC PowerMax 8000 array

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 Store data more efficiently and increase I/O performance with lower latency with a Dell EMC PowerMax 8000 array.

We concluded our hands-on testing on July 21, 2021. 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 June 17, 2021 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: Results of our testing.

	Dell EMC™ PowerMax 8000 solution	Vendor B solution
Data reduction		
Storage capacity needed for 64 TB of data (TB)	17.3	24.7
Data reduction ratio	3.6:1	2.5:1
Transactional database workload		
Max input/output per second (IOPS) with a simulated 32-thread online transaction processsing (OLTP) workload (Higher is better)	661,543	399,414
Max IOPS with a simulated 4-thread OLTP workload (Higher is better)	405,271	302,354
Latency, in milliseceonds (ms), with a simulated 4-thread OLTP workload (Lower is better)	0.621	0.839

	Dell EMC™ PowerMax 8000 solution	Vendor B solution
Data extraction workload		
Max IOPS with a simulated 64-thread data extraction workload (Higher is better)	616,719	169,255
Max throughput, in MB per second, with a simulated 64-thread data extraction workload (Higher is better)	30,112	8,264

System configuration information

Table 2: Detailed information on the servers we used in testing the two storage solutions.

System configuration information	4 x Dell EMC™ PowerEdge™ R740
BIOS name and version	Dell EMC PowerEdge R740 2.9.4
Non-default BIOS settings	Virtualization enabled
Operating system name and version/build number	VMware® ESXi™ 7.0.1 Update 1 Build 17325551
Date of last OS updates/patches applied	6/01/2021
Power management policy	Performance
Processor	
Number of processors	2
Vendor and model	Intel® Xeon® Gold 6240R
Core count (per processor)	24
Core frequency (GHz)	2.40
Memory module(s)	
Total memory in system (GB)	256
Number of memory modules	4
Vendor and model	Hynix® HMAA8GR7AJR4N-XN
Size (GB)	64
Type	PC4-2666
Speed (MHz)	2,933
Speed running in the server (MHz)	2,933
Storage controller	
Vendor and model	Dell PERC H330
Cache size (GB)	N/A
Firmware version	25.5.8.0001
Driver version	7.712.51.00
Local storage	
Number of drives	1
Drive vendor and model	Samsung® MZILT1T9HBJR0D3
Drive size (TB)	1.92
Drive information (speed, interface, type)	12Gbps, SAS, SSD
Network adapter	
Vendor and model	Broadcom® Gigabit Ethernet BCM5720
Number and type of ports	2 x 1Gb and 2 x 10Gb
Driver version	21.60.16

System configuration information	4 x Dell EMC™ PowerEdge™ R740
Storage adapter	
Vendor and model	Emulex LPe35002-M2-D
Number and type of ports	4 x two-port 32Gb Fibre Channel
Firmware version	03.03.37
Power supplies	
Vendor and model	Dell 0CMPGMA03
Number of power supplies	2
Wattage of each (W)	1,100

Table 3: Detailed information on the storage arrays we tested.

Storage configuration information	Dell EMC PowerMax 8000	Vendor B storage solution
Software version	5978.711.711	Firmware on a pre-configured array, current as of mid-March 2021.
Number of storage shelves	3	1
Total number of drives	34	36
Drive size (TB)	1.92	1.92

How we tested

During our testing, both the Dell EMC PowerMax 8000 array and the Vendor B array were located in an offsite data center lab. We performed all testing remotely after traveling to the lab to inspect the server clients, the network implementation, and the storage arrays. We had full control over and unfettered access to the testbeds. We used the same four Dell EMC PowerEdge R740 servers in each testbed, and changed the Fibre Channel zoning from one solution to the other when switching between testing each solution. Each server used VMware ESXi™ 7.0 U1 and dual-port 32Gb Fibre Channel adapters, grouped into a single VMware vCenter® 7.0 server.

Once we received IP addresses for all physical and virtual components, we verified that the configurations of both testbeds were identical where possible and as closely as possible where an identical configuration wasn't achievable. For example, the PowerMax 8000 had 34 disks and the Vendor B solution had 36 disks, but both arrays used a RAID 5 configuration with two hot spares. Both arrays were configured with the minimal number of supported storage shelves.

After completing the verification process, we moved into phase one of the three testing phases. In phase one, we measured the inline data reduction ratio of each storage array. We began by creating 64 1TB volumes (also called logical unit numbers or LUNs) on both arrays. After finishing the volume creation, we mapped the volumes from the PowerMax array, and then later the volumes from the Vendor B array, to the four PowerEdge R740 servers. Once we had mapped the volumes to the hosts, we added four (raw device mapping) RDM disks to each of the 16 virtual machines on each testbed.

We started our set of tests by using Vdbench to prefill the volumes with 64TB of data and to set an inline compression ratio of 2:1 and a deduplication ratio of 2:1. We measured the volume usage before and after prefilling, and then recorded the overall inline data reduction ratio of each array. We completed the data reduction tests two more times, using the median reduction ratio for each array in our report.

After we completed the data reduction testing phase, we moved into phase two of our testing. We first repeated the volume creation and mapping portion from phase one. We then prefilled the volumes, but for this second phase, we used a 2:1 compression ratio and a 1:1 deduplication ratio representative of real world database datasets. To precondition the arrays before testing, we executed a steady state workload targeting 200,000 IOPS, which ran for four hours and consisted of a mix of block sizes and read and write ratios, with 16 threads. After we completed the preconditioning cycle, we moved to the simulated OLTP test and again used a mix of block sizes and read and write ratios, but this time at several different thread counts to assess array performance.

We used the following OLTP I/O profile:

- 20 percent 8KB random read hit
- 45 percent 8KB random read miss
- 15 percent 8KB random write
- 10 percent 64KB sequential read
- 10 percent 64KB sequential write

The simulated OLTP I/O workload had a 90/10 skew, meaning the test presented 90 percent of the workload to 10 percent of the logical address space to mimic typical production environments. For phase three, using the same volumes from phase two, we ran a simulated extraction phase of a typical extract, transform, and load (ETL) workload. We again used a mix of block sizes and read and write ratios at multiple different thread counts.

We used the following data extraction profile:

- 5 percent 8KB random read hit
- 18 percent 8KB random read miss
- 2 percent 8KB random write
- 75 percent 64KB sequential read

The simulated data extraction I/O workload also had a 90/10 skew.

We ran a cache flush workload after the steady state preconditioning cycle and OLTP simulation tests to flush data from areas of the volume where the solution did not access or write data during the OLTP and data extraction test phases. This action removed test data from the cache.

We ran this series of tests three times in the following order and selected the median outputs of each solution to use in our report:

- Volume prefill
- Steady state
- Cache flush
- OLTP simulation
- Cache flush
- Data extraction

Testing inline data reduction

Prefilling the volumes with data

We used 128KB sequential writes with a single thread to fill the volumes with 64 TB of data. We performed this phase by running the following configuration, which ran on each of the 16 VMs:

```
messagescan=no
compratio=2
dedupratio=2
dedupunit=131072
dedupsets=5%
hd=default,vdbench=/bench/ptkit/vd,master=192.168.1.100,user=root,shell=ssh,jvms=1
hd=OL_001,system=192.168.1.101
...
hd=OL_016,system=192.168.1.116
sd=sd001,host=OL_001,lun=/dev/sdb,openflags=o_direct
sd=sd001,host=OL_001,lun=/dev/sdc,openflags=o_direct
sd=sd001,host=OL_001,lun=/dev/sdd,openflags=o_direct
sd=sd001,host=OL_001,lun=/dev/sde,openflags=o_direct
sd=sd016,host=OL_016,lun=/dev/sdb,openflags=o_direct
sd=sd016,host=OL_016,lun=/dev/sdc,openflags=o_direct
sd=sd016,host=OL_016,lun=/dev/sdd,openflags=o_direct
sd=sd016,host=OL_016,lun=/dev/sde,openflags=o_direct
wd=wd_MIGRATETS_SW,sd=*,seekpct=eof
rd=rd_MIGRATETS,wd=wd_MIGRATETS_
SW,elapsed=24h,interval=10,forxfersize=(128k),forrdpct=(0),forthreads=(1),iorate=max,maxdata=64000g
```

Testing simulated OLTP performance

Prefilling the volumes with data

We used 128KB sequential writes with a single thread to fill the volumes with data. We performed this phase by running the following configuration, which ran on each of the 16 VMs:

```
messagescan=no
compratio=2
hd=default,vdbench=/bench/ptkit/vd,master=192.168.1.100,user=root,shell=ssh,jvms=1
hd=OL_001,system=192.168.1.101
...
hd=OL_016,system=192.168.1.116
sd=sd001,host=OL_001,lun=/dev/sdb,openflags=o_direct
sd=sd001,host=OL_001,lun=/dev/sdc,openflags=o_direct
sd=sd001,host=OL_001,lun=/dev/sdd,openflags=o_direct
sd=sd001,host=OL_001,lun=/dev/sde,openflags=o_direct
sd=sd016,host=OL_016,lun=/dev/sdb,openflags=o_direct
sd=sd016,host=OL_016,lun=/dev/sdc,openflags=o_direct
sd=sd016,host=OL_016,lun=/dev/sdd,openflags=o_direct
sd=sd016,host=OL_016,lun=/dev/sde,openflags=o_direct
wd=wd_MIGRATETS_SW,sd=*,seekpct=eof
rd=rd_MIGRATETS,wd=wd_MIGRATETS_
SW,elapsed=24h,interval=10,forxfersize=(128k),forrdpct=(0),forthreads=(1),iorate=max,maxdata=64000g
```

Getting the solutions to a steady state

We deployed 64 1TB volumes and configured Vdbench to run a workload that emulates a typical OLTP workload at a steady state of 200,000 IOPS. We performed this phase by running the following configuration, which ran on each of the 16 VMs:

```
messagescan=no
compratio=2
hd=default,vdbench=/bench/ptkit/vd,master=192.168.1.100,user=root,shell=ssh,jvms=1
hd=OL_001,system=192.168.1.101
...
hd=OL_016,system=192.168.1.116
sd=sd001,host=OL_001,lun=/dev/sdb,openflags=o_direct
sd=sd001,host=OL_001,lun=/dev/sdc,openflags=o_direct
sd=sd001,host=OL_001,lun=/dev/sdd,openflags=o_direct
sd=sd001,host=OL_001,lun=/dev/sde,openflags=o_direct
sd=sd016,host=OL_016,lun=/dev/sdb,openflags=o_direct
sd=sd016,host=OL_016,lun=/dev/sdc,openflags=o_direct
sd=sd016,host=OL_016,lun=/dev/sdd,openflags=o_direct
sd=sd016,host=OL_016,lun=/dev/sde,openflags=o_direct
wd=wd_STEADYSTATETS_RRH,sd=*,rhpct=100,rdpct=100,xfersize=8K,skew=20,range=(10m,30m)
wd=wd_STEADYSTATETS_RM1,sd=*,rdpct=100,xfersize=8k,skew=40,range=(89,99)
wd=wd_STEADYSTATETS_RM2,sd=*,rdpct=100,xfersize=8k,skew=5,range=(11,88)
wd=wd_STEADYSTATETS_RW1,sd=*,rdpct=0,xfersize=8K,skew=13,range=(89,99)
wd=wd_STEADYSTATETS_RW2,sd=*,rdpct=0,xfersize=8K,skew=2,range=(11,88)
wd=wd_STEADYSTATETS_SR1,sd=*,rdpct=100,seekpct=seqnz,range=(89,99),xfersize=64K,skew=9
wd=wd_STEADYSTATETS_SR2,sd=*,rdpct=100,seekpct=seqnz,range=(11,88),xfersize=64K,skew=1
wd=wd_STEADYSTATETS_SW1,sd=*,rdpct=0,seekpct=seqnz,range=(89,99),xfersize=64K,skew=9
wd=wd_STEADYSTATETS_SW2,sd=*,rdpct=0,seekpct=seqnz,range=(11,88),xfersize=64K,skew=1
rd=rd_STEADYSTATETS,wd=wd_
    STEADYSTATETS_*,iorate=200000,elapsed=4h,interval=10,warmup=60,forthreads=(16),hitarea=6m
```

Running the simulated OLTP workload

For these tests, we deployed 64 1TB volumes and configured Vdbench to run a workload that emulates a typical OLTP workload with 4 and 32 threads. We performed this phase by running the following configuration, which ran on each of the 16 VMs:

```
messagescan=no
compratio=2
hd=default,vdbench=/bench/ptkit/vd,master=192.168.1.100,user=root,shell=ssh,jvms=1
hd=OL_001,system=192.168.1.101
...
hd=OL_016,system=192.168.1.116
sd=sd001,host=OL_001,lun=/dev/sdb,openflags=o_direct
sd=sd001,host=OL_001,lun=/dev/sdc,openflags=o_direct
sd=sd001,host=OL_001,lun=/dev/sdd,openflags=o_direct
sd=sd001,host=OL_001,lun=/dev/sde,openflags=o_direct
sd=sd016,host=OL_016,lun=/dev/sdb,openflags=o_direct
sd=sd016,host=OL_016,lun=/dev/sdc,openflags=o_direct
sd=sd016,host=OL_016,lun=/dev/sdd,openflags=o_direct
sd=sd016,host=OL_016,lun=/dev/sde,openflags=o_direct
wd=wd_OLTP2S_RRH,sd=*,rhpct=100,rdpct=100,xfersize=8K,skew=20,range=(10m,30m)
wd=wd_OLTP2S_RM1,sd=*,rdpct=100,xfersize=8k,skew=40,range=(89,99)
wd=wd_OLTP2S_RM2,sd=*,rdpct=100,xfersize=8k,skew=5,range=(11,88)
wd=wd_OLTP2S_RW1,sd=*,rdpct=0,xfersize=8K,skew=13,range=(89,99)
wd=wd_OLTP2S_RW2,sd=*,rdpct=0,xfersize=8K,skew=2,range=(11,88)
wd=wd_OLTP2S_SR1,sd=*,rdpct=100,seekpct=seqnz,range=(89,99),xfersize=64K,skew=9
wd=wd_OLTP2S_SR2,sd=*,rdpct=100,seekpct=seqnz,range=(11,88),xfersize=64K,skew=1
wd=wd_OLTP2S_SW1,sd=*,rdpct=0,seekpct=seqnz,range=(89,99),xfersize=64K,skew=9
wd=wd_OLTP2S_SW2,sd=*,rdpct=0,seekpct=seqnz,range=(11,88),xfersize=64K,skew=1
rd=rd_OLTP2S,wd=wd_OLTP2S_*,iorate=max,elapsed=120,interval=10,warmup=60,forthreads=(4,32),hitarea=6m
```

Testing data extraction for a simulated ETL process

For these tests, we used the same 64 1TB volumes from the previous test phase and configured Vdbench to run a simulated extraction phase that you would see in an ETL workload. We used multiple block sizes and ran the test with 64 threads. We performed this phase by running the following configuration, which ran on each of the 16 VMs:

```
messagescan=no
compratio=2
hd=default,vdbench=/bench/ptkit/vd,master=192.168.1.100,user=root,shell=ssh,jvms=1
hd=OL_001,system=192.168.1.101
...
hd=OL_016,system=192.168.1.116
sd=sd001,host=OL_001,lun=/dev/sdb,openflags=o_direct
sd=sd001,host=OL_001,lun=/dev/sdc,openflags=o_direct
sd=sd001,host=OL_001,lun=/dev/sdd,openflags=o_direct
sd=sd001,host=OL_001,lun=/dev/sde,openflags=o_direct
sd=sd016,host=OL_016,lun=/dev/sdb,openflags=o_direct
sd=sd016,host=OL_016,lun=/dev/sdc,openflags=o_direct
sd=sd016,host=OL_016,lun=/dev/sdd,openflags=o_direct
sd=sd016,host=OL_016,lun=/dev/sde,openflags=o_direct
wd=wd_ETLS_RRH,sd=*,rhpct=100,rdpct=100,xfersize=8K,skew=5,range=(10m,30m)
wd=wd_ETLS_RW1,sd=*,rdpct=0,xfersize=8K,skew=2,range=(89,99)
wd=wd_ETLS_RM1,sd=*,rdpct=100,xfersize=8k,skew=16,range=(89,99)
wd=wd_ETLS_RM2,sd=*,rdpct=100,xfersize=8k,skew=2,range=(11,88)
wd=wd_ETLS_SR1,sd=*,rdpct=100,seekpct=seqnz,range=(89,99),xfersize=64K,skew=66
wd=wd_ETLS_SR2,sd=*,rdpct=100,seekpct=seqnz,range=(11,88),xfersize=64K,skew=9
rd=rd_ETLS,wd=wd_ETLS_*,iorate=max,elapsed=120,interval=10,warmup=60,forthreads=(64),hitarea=6m
```

Flushing the cache

For these tests, we deployed 64 1TB volumes and configured Vdbench to run 128KB sequential reads to the cache with one thread. We performed this phase by running the following configuration, which ran on each of the 16 VMs:

```
messagescan=no
compratio=2
hd=default,vdbench=/bench/ptkit/vd,master=192.168.1.100,user=root,shell=ssh
hd=OL_001,system=192.168.1.101
...
hd=OL_016,system=192.168.1.116
sd=sd001,host=OL_001,lun=/dev/sdb,openflags=o_direct
sd=sd001,host=OL_001,lun=/dev/sdc,openflags=o_direct
sd=sd001,host=OL_001,lun=/dev/sdd,openflags=o_direct
sd=sd001,host=OL_001,lun=/dev/sde,openflags=o_direct
sd=sd016,host=OL_016,lun=/dev/sdb,openflags=o_direct
sd=sd016,host=OL_016,lun=/dev/sdc,openflags=o_direct
sd=sd016,host=OL_016,lun=/dev/sdd,openflags=o_direct
sd=sd016,host=OL_016,lun=/dev/sde,openflags=o_direct
wd=wd_CACHEFLUSH_RRM,sd=*,rdpct=100,xfersize=128k,range=(1,10)

rd=rd_CACHEFLUSH,wd=wd_CACHEFLUSH_RRM,forthreads=(1),iorate=max,elapsed=900,interval=10,maxdata=4096g
```

Read the report at http://facts.pt/wAiaXaY ▶

This project was commissioned by Dell Technologies.

Principled Technologies®

Facts matter.®