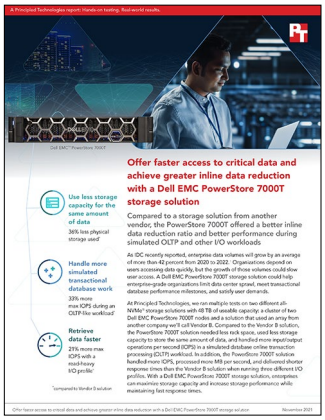

The science behind the report:

Offer faster access to critical data and achieve greater inline data reduction with a Dell EMC PowerStore 7000T storage solution

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 Offer faster access to critical data and achieve greater inline data reduction with a Dell EMC PowerStore 7000T storage solution.

We concluded our hands-on testing on August 27, 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 August 10, 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™ PowerStore 7000T solution	Vendor B solution
Data reduction		
Storage capacity needed for 16 TB of data (TB)	3.90	6.15
Data reduction ratio	4.2:1	2.6:1
32-thread simulated online transaction processing (OLTP) workload		
Input/output operations per second (IOPS)	534,092	400,859
Read/write workloads		
Max IOPS with 8KB 32-thread random read workload (IOPS)	1,231,617	1,012,745
Max IOPS with 4KB 32-thread random write workload (IOPS)	800,478	582,119
Max throughput with 256KB 32-thread sequential read workload (MB/s)	48,850	9,285

	Dell EMC™ PowerStore 7000T solution	Vendor B solution
IOPS with 32KB 4-thread 70 percent read workload (IOPS)	308,984	236,599
Latency with 32KB 4-thread 70 percent read workload (ms)	0.825	1.078

System configuration information

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

System configuration information	4 x Dell EMC™ PowerEdge™ R740
BIOS name and version	Dell EMC PowerEdge R740 2.11
Non-default BIOS settings	Virtualization enabled
Operating system name and version/build number	VMware® ESXi™ 7.0.1 Update 2 Build 17867351
Date of last OS updates/patches applied	7/12/2021
Power management policy	Performance
Processor	
Number of processors	2
Vendor and model	Intel® Xeon® Gold 6126
Core count (per processor)	12
Core frequency (GHz)	2.60
Memory module(s)	
Total memory in system (GB)	256
Number of memory modules	8
Vendor and model	Samsung® M393A2K43BB1-CTD
Size (GB)	32
Type	PC4-2666
Speed (MHz)	2,666
Speed running in the server (MHz)	2,666
Local storage	
Number of drives	1
Drive vendor and model	Samsung® MZ7LH240HAHQ0D3
Drive size (GB)	240
Drive information (speed, interface, type)	6Gbps, SAS, SSD
Network adapter	
Vendor and model	Broadcom® Gigabit Ethernet BCM5720
Number and type of ports	2 x 1Gb, 2 x 10Gb
Driver version	21.40.9
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

System configuration information	4 x Dell EMC™ PowerEdge™ R740
Power supplies	
Vendor and model	Dell 0CMPGMA02
Number of power supplies	2
Wattage of each (W)	1,100

Table 3: Detailed information on the storage we tested.

Storage configuration information	Dell EMC PowerStore 7000T solution	Vendor B storage solution
Software version	2.0.0.0	Firmware on a pre-configured array, current as of mid-March 2021
Number of storage shelves	2	1
Total number of drives	36	36
Drive size (TB)	1.92	1.92

How we tested

During our testing, both the Dell EMC PowerStore 7000T solution and the Vendor B solution 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 test beds. We used the same four Dell EMC PowerEdge R740 servers in each test bed, with VMware ESXi™ 7.0 U2 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 test beds were identical where possible and as close as possible where an identical configuration wasn't achievable.

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 solution. We began by creating 16 1TB volumes (also called logical unit numbers or LUNs) on both solutions. After finishing the volume creation, we mapped the volumes from the PowerStore and Vendor B solutions to the PowerEdge R740 servers. Once we mapped the volumes to the hosts, we added two raw device mapping (RDM) disks to each of the eight virtual machines on each test bed.

We started our set of tests by using Vdbench to prefill the volumes with 16 TB of data and to set an inline compression ratio of 2:1 and a deduplication ratio of 2:1. We measured the system physical usage immediately before and after prefilling, and then recorded the overall inline data reduction ratio of each solution. We completed the data reduction tests two more times, using the median reduction ratio in our report for each solution.

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, except for this phase, we created 64 1TB volumes and mapped them to 16 VMs. We then prefilled the volumes, but for this second phase, we used a 2:1 compression ratio and a 1:1 deduplication ratio. To precondition the solutions 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 (listed below), but this time at several different thread counts to assess solution 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, we again created 64 1TB volumes and mapped them to 16 VMs. We repeated the same prefill process from phase two, but for this third phase, we used a 2:1 compression ratio and a 2:1 deduplication ratio. We then used Vdbench to run several synthetic I/O workloads (listed below) on the dataset.

We used the following profiles:

- 8KB random read
- 32KB random 70 percent read
- 4KB random write
- 256KB sequential read

The synthetic I/O workloads also had a 90/10 skew. We ran a cache flush workload prior to each workload to avoid a potential cache hit.

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

a. Volume prefill
b. Steady state
c. Cache flush
d. OLTP simulation

We ran the synthetic I/O portion of the tests three times in the following order and selected the median outputs of each solution:

a. Volume prefill
b. Steady state
c. Cache flush
d. 8KB random read
e. Cache flush
f. 256KB sequential read
g. Cache flush
h. 32KB random 70 percent read
i. Cache flush
j. 4KB random writes

Testing inline data reduction

Prefilling the volumes with data

We used 128KB sequential writes with a single thread to fill the volumes with 16 TB of data. We performed this phase by running the following configuration, which ran on each of the 8 VMs:

```
messagescan=no
compratio=2
dedupratio=2
dedupunit=(For Vendor B, this value was 4096; for PowerStore, this value was 8192)
dedupsets=5%
hd=default,vdbench=/bench/ptkit/vd,master=192.168.1.200,user=root,shell=ssh,jvms=1
hd=PS_001,system=192.168.1.201
...
hd=PS_008,system=192.168.1.208
sd=sd001,host=PS_001,lun=/dev/sdb,openflags=o_direct
sd=sd002,host=PS_001,lun=/dev/sdc,openflags=o_direct
...
sd=sd015,host=PS_008,lun=/dev/sdb,openflags=o_direct
sd=sd016,host=PS_008,lun=/dev/sdc,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=192.168.1.200,user=root,shell=ssh,jvms=1
hd=PS_001,system=192.168.1.201
...
hd=PS_016,system=192.168.1.216
sd=sd001,host=PS_001,lun=/dev/sdb,openflags=o_direct
sd=sd002,host=PS_001,lun=/dev/sdc,openflags=o_direct
sd=sd003,host=PS_001,lun=/dev/sdd,openflags=o_direct
sd=sd004,host=PS_001,lun=/dev/sde,openflags=o_direct
...
sd=sd061,host=PS_016,lun=/dev/sdb,openflags=o_direct
sd=sd062,host=PS_016,lun=/dev/sdc,openflags=o_direct
sd=sd063,host=PS_016,lun=/dev/sdd,openflags=o_direct
sd=sd064,host=PS_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=192.168.1.200,user=root,shell=ssh,jvms=1
hd=PS_001,system=192.168.1.201
...
hd=PS_016,system=192.168.1.216
sd=sd001,host=PS_001,lun=/dev/sdb,openflags=o_direct
sd=sd002,host=PS_001,lun=/dev/sdc,openflags=o_direct
sd=sd003,host=PS_001,lun=/dev/sdd,openflags=o_direct
sd=sd004,host=PS_001,lun=/dev/sde,openflags=o_direct
...
sd=sd061,host=PS_016,lun=/dev/sdb,openflags=o_direct
sd=sd062,host=PS_016,lun=/dev/sdc,openflags=o_direct
sd=sd063,host=PS_016,lun=/dev/sdd,openflags=o_direct
sd=sd064,host=PS_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 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=192.168.1.200,user=root,shell=ssh,jvms=1
hd=PS_001,system=192.168.1.201
...
hd=PS_016,system=192.168.1.216
sd=sd001,host=PS_001,lun=/dev/sdb,openflags=o_direct
sd=sd002,host=PS_001,lun=/dev/sdc,openflags=o_direct
sd=sd003,host=PS_001,lun=/dev/sdd,openflags=o_direct
sd=sd004,host=PS_001,lun=/dev/sde,openflags=o_direct
...
sd=sd061,host=PS_016,lun=/dev/sdb,openflags=o_direct
sd=sd062,host=PS_016,lun=/dev/sdc,openflags=o_direct
sd=sd063,host=PS_016,lun=/dev/sdd,openflags=o_direct
sd=sd064,host=PS_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=(32),hitarea=6m
```

Testing synthetic I/O workloads

Prefilling the volumes with data

We used 128KB sequential writes with a single thread to fill the volumes with 64 TB of data and 2:1 compression and 2:1 deduplication ratios. We performed this phase by running the following configuration, which ran on each of the 16 VMs:

```
messagescan=no
compratio=2
dedupratio=2
dedupunit=8192
dedupsets=5%
hd=default,vdbench=/bench/ptkit/vd,master=192.168.1.200,user=root,shell=ssh,jvms=1
hd=PS_001,system=192.168.1.201
...
hd=PS_016,system=192.168.1.216
sd=sd001,host=PS_001,lun=/dev/sdb,openflags=o_direct
sd=sd002,host=PS_001,lun=/dev/sdc,openflags=o_direct
sd=sd003,host=PS_001,lun=/dev/sdd,openflags=o_direct
sd=sd004,host=PS_001,lun=/dev/sde,openflags=o_direct
...
sd=sd061,host=PS_016,lun=/dev/sdb,openflags=o_direct
sd=sd062,host=PS_016,lun=/dev/sdc,openflags=o_direct
sd=sd063,host=PS_016,lun=/dev/sdd,openflags=o_direct
sd=sd064,host=PS_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

For these tests, we deployed 64 1TB LUNs 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
dedupratio=2
dedupunit=8192
dedupsets=5%
d=default,vdbench=/bench/ptkit/vd,master=192.168.1.200,user=root,shell=ssh,jvms=1
hd=PM_001,system=192.168.1.201
...
hd=PM_016,system=192.168.1.216
sd=sd_001,host=PM_001,lun=/dev/sdb,openflags=o_direct
sd=sd_002,host=PM_001,lun=/dev/sdc,openflags=o_direct
sd=sd_003,host=PM_001,lun=/dev/sdd,openflags=o_direct
sd=sd_004,host=PM_001,lun=/dev/sde,openflags=o_direct
...
sd=sd_061,host=PM_016,lun=/dev/sdb,openflags=o_direct
sd=sd_062,host=PM_016,lun=/dev/sdd,openflags=o_direct
sd=sd_063,host=PM_016,lun=/dev/sdc,openflags=o_direct
sd=sd_064,host=PM_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 8KB random read workload

We used a 8KB block size and 100 percent random reads with 32 threads running against 64TB of data, a 2:1 compression ratio, and a 2:1 deduplication ratio. We performed this phase by running the following configuration, which ran on each of the 16 VMs:

```
messagescan=no
compratio=2
dedupratio=2
dedupunit=8192
dedupsets=5%
hd=default,vdbench=/bench/ptkit/vd,master=192.168.1.200,user=root,shell=ssh
hd=PS_001,system=192.168.1.201
...
hd=PS_016,system=192.168.1.216
sd=sd001,host=PS_001,lun=/dev/sdb,openflags=o_direct
sd=sd002,host=PS_001,lun=/dev/sdc,openflags=o_direct
sd=sd003,host=PS_001,lun=/dev/sdd,openflags=o_direct
sd=sd004,host=PS_001,lun=/dev/sde,openflags=o_direct
...
sd=sd061,host=PS_016,lun=/dev/sdb,openflags=o_direct
sd=sd062,host=PS_016,lun=/dev/sdc,openflags=o_direct
sd=sd063,host=PS_016,lun=/dev/sdd,openflags=o_direct
sd=sd064,host=PS_016,lun=/dev/sde,openflags=o_direct
wd=wd_RRM8K_RM,sd=*,rdpct=100,range=(11,99),xfersize=8k
rd=rd_RRM8K,wd=wd_RRM8K_RM,iorate=max,elapsed=120,interval=10,warmup=60,forthreads=(32)
```

Running the 256KB sequential read workload

We used a 256KB block size and 100 percent sequential reads with 32 threads running against 64 TB of data, a 2:1 compression ratio, and a 2:1 deduplication ratio. We performed this phase by running the following configuration, which ran on each of the 16 VMs:

```
messagescan=no
compratio=2
dedupratio=2
dedupunit=8192
dedupsets=5%
hd=default,vdbench=/bench/ptkit/vd,master=192.168.1.200,user=root,shell=ssh
hd=PS_001,system=192.168.1.201
...
hd=PS_016,system=192.168.1.216
sd=sd001,host=PS_001,lun=/dev/sdb,openflags=o_direct
sd=sd002,host=PS_001,lun=/dev/sdc,openflags=o_direct
sd=sd003,host=PS_001,lun=/dev/sdd,openflags=o_direct
sd=sd004,host=PS_001,lun=/dev/sde,openflags=o_direct
...
sd=sd061,host=PS_016,lun=/dev/sdb,openflags=o_direct
sd=sd062,host=PS_016,lun=/dev/sdc,openflags=o_direct
sd=sd063,host=PS_016,lun=/dev/sdd,openflags=o_direct
sd=sd064,host=PS_016,lun=/dev/sde,openflags=o_direct
wd=wd_SR256K_SR,sd=*,rdpct=100,seekpct=seqnz,range=(11,99),xfersize=256K
rd=rd_SR256K,wd=wd_SR256K_SR,iorate=max,elapsed=120,interval=10,warmup=60,forthreads=(32)
```

Running the 32KB random 70 percent read workload

We used a 32KB block size and 70 percent random reads with 4 threads running against 64 TB of data, a 2:1 compression ratio, and a 2:1 deduplication ratio. We performed this phase by running the following configuration, which ran on each of the 16 VMs:

```
messagescan=no
compratio=2
dedupratio=2
dedupunit=8192
dedupsets=5%
hd=default,vdbench=/bench/ptkit/vd,master=192.168.1.200,user=root,shell=ssh
hd=PS_001,system=192.168.1.201
...
hd=PS_016,system=192.168.1.216
sd=sd001,host=PS_001,lun=/dev/sdb,openflags=o_direct
sd=sd002,host=PS_001,lun=/dev/sdc,openflags=o_direct
sd=sd003,host=PS_001,lun=/dev/sdd,openflags=o_direct
sd=sd004,host=PS_001,lun=/dev/sde,openflags=o_direct
...
sd=sd061,host=PS_016,lun=/dev/sdb,openflags=o_direct
sd=sd062,host=PS_016,lun=/dev/sdc,openflags=o_direct
sd=sd063,host=PS_016,lun=/dev/sdd,openflags=o_direct
sd=sd064,host=PS_016,lun=/dev/sde,openflags=o_direct
wd=wd_RW703032K_ALL,sd=*,xfersize=32k,seekpct=100,rdpct=70,range=(11,99)
rd=rd_RW703032K,wd=wd_RW703032K_ALL,iorate=max,elapsed=120,interval=10,warmup=60,forthreads=(4)
```

Running the 4KB random write workload

We used a 4KB block size and 100 percent random writes with 32 threads running against 64 TB of data, a 2:1 compression ratio, and a 2:1 deduplication ratio. We performed this phase by running the following configuration, which ran on each of the 16 VMs:

```
messagescan=no
compratio=2
dedupratio=2
dedupunit=8192
dedupsets=5%
hd=default,vdbench=/bench/ptkit/vd,master=192.168.1.200,user=root,shell=ssh
hd=PS_001,system=192.168.1.201
...
hd=PS_016,system=192.168.1.216
sd=sd001,host=PS_001,lun=/dev/sdb,openflags=o_direct
sd=sd002,host=PS_001,lun=/dev/sdc,openflags=o_direct
sd=sd003,host=PS_001,lun=/dev/sdd,openflags=o_direct
sd=sd004,host=PS_001,lun=/dev/sde,openflags=o_direct
...
sd=sd061,host=PS_016,lun=/dev/sdb,openflags=o_direct
sd=sd062,host=PS_016,lun=/dev/sdc,openflags=o_direct
sd=sd063,host=PS_016,lun=/dev/sdd,openflags=o_direct
sd=sd064,host=PS_016,lun=/dev/sde,openflags=o_direct
wd=wd_RWM4K_RW,sd=*,rdpct=0,seekpct=100,range=(11,99),xfersize=4K
rd=rd_RWM4K,wd=wd_RWM4K_RW,iorate=max,elapsed=120,interval=10,warmup=60,forthreads=(32)
```

Flushing the cache

For phase two 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=192.168.21.200,user=root,shell=ssh
hd=PS_001,system=192.168.21.201
...
hd=PS_016,system=192.168.21.216
sd=sd001,host=PSOL_001,lun=/dev/sdb,openflags=o_direct
sd=sd002,host=PS_001,lun=/dev/sdc,openflags=o_direct
sd=sd003,host=PS_001,lun=/dev/sdd,openflags=o_direct
sd=sd004,host=PS_001,lun=/dev/sde,openflags=o_direct
...
sd=sd061,host=PS_016,lun=/dev/sdb,openflags=o_direct
sd=sd062,host=PS_016,lun=/dev/sdc,openflags=o_direct
sd=sd063,host=PS_016,lun=/dev/sdd,openflags=o_direct
sd=sd064,host=PS_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
```

For phase three 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
dedupratio=2
dedupunit=8192
dedupsets=5%
d=default,vdbench=/bench/ptkit/vd,master=192.168.1.200,user=root,shell=ssh,jvms=1
hd=PM_001,system=192.168.1.201
...
hd=PM_016,system=192.168.1.216
sd=sd_001,host=PM_001,lun=/dev/sdb,openflags=o_direct
sd=sd_002,host=PM_001,lun=/dev/sdc,openflags=o_direct
sd=sd_003,host=PM_001,lun=/dev/sdd,openflags=o_direct
sd=sd_004,host=PM_001,lun=/dev/sde,openflags=o_direct
...
sd=sd_061,host=PM_016,lun=/dev/sdb,openflags=o_direct
sd=sd_062,host=PM_016,lun=/dev/sdd,openflags=o_direct
sd=sd_063,host=PM_016,lun=/dev/sdc,openflags=o_direct
sd=sd_064,host=PM_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/pemecAX ▶

This project was commissioned by Dell Technologies.

Principled Technologies®

Facts matter.®