



The science behind the report:

Get more I/O performance on fewer servers with Intel Optane DC SSDs

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 [Get more I/O performance on fewer servers with Intel Optane DC SSDs](#).

We concluded our hands-on testing on August 7, 2019. 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 2, 2019 or earlier. Unavoidably, these configurations may not represent the latest versions available when this report appears.

Our results

The table below presents our findings in detail. We tested each configuration in HPE ProLiant DL380 Gen10 servers, using a 70/30% read/write workload with 8KB block sizes.

Input/output operations per second (IOPS)

Configuration with SATA SSDs	Configuration with SATA SSDs + Intel® Optane™ DC SSDs
387,092	592,173
	52.9% win

Latency (ms)

Configuration with SATA SSDs	Configuration with SATA SSDs + Intel® Optane™ DC SSDs
6.0	4.4
	26.6% win

System configuration information

The table below presents detailed information on the systems we tested. Both configurations used HPE ProLiant DL380 Gen10 servers. The first configuration used four SATA SSDs in the capacity tier. The second configuration used four SATA SSDs in the capacity tier plus two Intel Optane DC SSDs in the caching tier.

Server configuration information	HPE ProLiant DL380 Gen10 servers
Tested by	Principled Technologies
Test date	08/07/2019
Workload & version	VMFleet/DISKSPD 2.0.21a
Server platform	HPE ProLiant DL380 Gen10
BIOS name and version	U30 v1.46 (10/02/2018)
BIOS settings	Performance
Operating system name and version/build number	Windows Server 2019 Build 1809
Date of last OS updates/patches applied	08/02/2019
Power management policy	Static High Performance Mode
Processor	
Number of processors	2
Vendor and model	Intel Xeon® Gold 6154 CPU @ 3.00GHz
Core count (per processor)	18
Core frequency (GHz)	3.0
Stepping	H0
Hyper-Threading	Yes
Turbo	Yes
Memory module(s)	
Total memory in system (GB)	384
Number of memory modules	12
Vendor and model	Samsung® M393A4K40BB2-CTD6Q
Size (GB)	32
Type	PC4-2666V
Speed (MHz)	2,666
Speed running in the server (MHz)	2,666
NVMe memory present?	No
Total memory (DDR+NVMe RAM)	384
Local storage (OS)	
Number of drives	1
Drive vendor and model	Intel SSD DC S3700
Drive size (GB)	400
Drive information (speed, interface, type)	SATA 6GB/s SSD

Server configuration information		HPE ProLiant DL380 Gen10 servers
Local storage (SATA SSD)		
Number of drives	4	
Drive vendor and model	HPE 3.84TB SATA RI SFF SC DS SSD (Intel SSD D3-S4510)	
Drive size (TB)	3.84	
Drive information (speed, interface, type)	SATA 6Gb/s SSD	
Local storage (NVMe SSD)		
Number of drives	2	
Drive vendor and model	HPE 375GB NVMe x4 WI SFF SCN DS SSD (Intel Optane SSD DC P4800X)	
Drive size (GB)	375	
Drive information (interface, type)	Intel Optane DC SSD	
Network adapter		
Vendor and model	Intel XXV710	
Number and type of ports	2 x 25GbE	
Cooling fans		
Vendor and model	Delta PFM0612XHE	
Number of cooling fans	6	
Power supplies		
Vendor and model	HPE 865414-B21	
Number of power supplies	2	
Wattage of each (W)	800	

How we tested

Configuring the infrastructure server

We used a 2U, dual-socket server to host the Active Directory and Storage Spaces Direct management VMs. We installed Windows Server 2019 Datacenter edition and added the Hyper-V role. On the Active Directory VM, we added the Active Directory Domain Service and DNS Server roles and created a new domain for our test cluster. On the Storage Spaces Direct management VM, we added Remote Server Administration Tools (RSAT) for PowerShell and Failover Clustering.

Configuring the cluster nodes

We used four HPE ProLiant DL380 Gen10 servers for our test cluster. We set the energy and cooling profiles on each server to Performance mode. We installed Windows Server 2019 Datacenter Edition onto a 128GB SATA SSD, separate from the drives designated for cluster storage. We then set the power policy to High Performance, configured the network adapters and joined each server to the domain. Finally, we ran the network and storage commands to enable Storage Spaces Direct. We performed the following steps for each server:

Installing Windows Server 2019

1. Boot the server to the Windows Server 2019 installation media.
2. At the prompt, press any key to boot from the CD/DVD location.
3. Click Next.
4. Click Install Now.
5. Select Windows Server 2019 Datacenter Edition (Desktop Experience), and click Next.
6. Check I accept the license terms, and click Next.
7. Select the 128GB OS drive, and click Next.
8. After installation, enter a password for Administrator, and click Finish.
9. Open Control Panel and search for Power Options.
10. Click Power Options.
11. Change the power policy to High Performance, and click Apply.
12. Close the window.
13. Update the HP system and Intel network drivers.
14. Run the following commands to configure NUMA addressing for the network cards (the syntax will vary based on the card manufacturer and physical placement):

```
$targets = "s2dnode1","s2dnode2","s2dnode3","s2dnode4"
```

```
invoke-command -ComputerName $targets -ScriptBlock {set-netadapterrss -Name "NIC1"  
-BaseProcessorGroup 0 -MaxProcessorGroup 0 -BaseProcessorNumber 2 -MaxProcessorNumber 19  
-MaxProcessors 16 -NumberOfReceiveQueues 8 -NumaNode 0 -Profile NUMA }
```

```
start-sleep 120
```

```
invoke-command -ComputerName $targets -ScriptBlock { set-netadapterrss -Name "NIC2"  
-BaseProcessorGroup 0 -MaxProcessorGroup 0 -BaseProcessorNumber 20 -MaxProcessorNumber 72  
-MaxProcessors 16 -NumberOfReceiveQueues 8 -NumaNode 0 -Profile NUMA }
```

Joining the server to the domain

1. Use the Windows Run dialog to open sysadm.cpl.
2. Click Change.
3. Select Domain, and enter the domain name.
4. Click OK.
5. Enter the password for the domain administrator, and click OK.
6. Reboot the server, and log in as the domain administrator.

Installing server roles and features

1. Open a PowerShell window as the domain administrator, and run the following command to install the required roles and features for Storage Spaces Direct:

```
Install-WindowsFeature -Name "Data-Center-Bridging","Failover-Clustering","Hyper-V","RSAT-Clustering-  
PowerShell","Hyper-V-PowerShell"
```

Running the cluster validation tool

1. Open a PowerShell window as the domain administrator.
2. To validate the cluster configuration, run the following command:

```
Test-Cluster -Node <Node1, Node2, Node3, Node4 > -Include "Storage Spaces Direct", "Inventory", "Network", "System Configuration"
```
3. Open the cluster validation report, and ensure there are no failures. If there are warnings in the report, verify that they are negligible, or correct the warnings before deployment.

Deploying Storage Spaces Direct cluster

1. Open a PowerShell window as the domain administrator.
2. Run the following command to create a new Storage Spaces Direct cluster without storage (you will add storage later):

```
New-Cluster -Name S2D2 -Node s2dnode1,s2dnode2,s2dnode3,s2dnode4 -NoStorage
```
3. Set the cluster quorum location to a file share on the Active Directory server:

```
Set-ClusterQuorum -FileShareWitness \\AD\ClusterQuorum
```

Enabling Storage Spaces Direct

1. Open a PowerShell window as the domain administrator.
2. Run the following command to enable Storage Spaces Direct:

```
Enable-ClusterS2D
```

Creating the volumes

1. Open a PowerShell window as the domain administrator.
2. Run the following commands to create a 4TB volume on each node, as well as an additional 100GB volume for storing VMFLEET scripts:
3. Create volumes for VMFleet:

```
New-Volume -FriendlyName "s2dnode1" -FileSystem CSVFS_ReFS -StoragePoolFriendlyName "**S2D*" -Size 4TB -ResiliencySettingName Mirror  
New-Volume -FriendlyName "s2dnode2" -FileSystem CSVFS_ReFS -StoragePoolFriendlyName "**S2D*" -Size 4TB -ResiliencySettingName Mirror  
New-Volume -FriendlyName "s2dnode3" -FileSystem CSVFS_ReFS -StoragePoolFriendlyName "**S2D*" -Size 4TB -ResiliencySettingName Mirror  
New-Volume -FriendlyName "s2dnode4" -FileSystem CSVFS_ReFS -StoragePoolFriendlyName "**S2D*" -Size 4TB -ResiliencySettingName Mirror  
New-Volume -FriendlyName "collect" -FileSystem CSVFS_ReFS -StoragePoolFriendlyName "**S2D*" -Size 100GB -ResiliencySettingName Mirror
```
4. Turn off cluster performance history logging:

```
Stop-ClusterPerformanceHistory -DeleteHistory
```
5. Run the following commands to assign the volumes to the correct hosts:

```
Move-ClusterSharedVolume -Name "Cluster Virtual Disk (collect)" -Node s2dnode1  
Move-ClusterSharedVolume -Name "Cluster Virtual Disk (s2dnode1)" -Node s2dnode1  
Move-ClusterSharedVolume -Name "Cluster Virtual Disk (s2dnode2)" -Node s2dnode2  
Move-ClusterSharedVolume -Name "Cluster Virtual Disk (s2dnode3)" -Node s2dnode3  
Move-ClusterSharedVolume -Name "Cluster Virtual Disk (s2dnode4)" -Node s2dnode4
```
6. Optimize the storage pool:

```
Optimize-StoragePool
```
7. Disable CSV cache, due to performance issues with VMfleet:

```
(Get-Cluster).BlockCacheSize = 0  
Get-ClusterSharedVolume | ForEach {  
    $Owner = $_.OwnerNode  
    $_ | Move-ClusterSharedVolume  
    $_ | Move-ClusterSharedVolume -Node $Owner  
}
```

Configuring VMFleet

VMFleet is a set of scripts designed to make it easy to clone out VMs and run Microsoft DiskSpd on each. We used Windows Server 2019 Datacenter Edition Server Core as our master image operating system. We used the VMFleet scripts and documentation at this GitHub repository: <https://github.com/Microsoft/diskspd/tree/master/Frameworks/VMFleet>.

Creating the master image

1. On one of the cluster nodes, open Hyper-V Manager and click New→Virtual Machine.
2. Click Next.
3. Select Generation 2, and click Next.
4. Click Next.
5. Click Next.
6. Select Create a virtual hard disk. Set the size to 32GB, and name it VMFleet_gold.vhdx
7. Click Next.
8. Click Finish.
9. Right-click the virtual machine, and click Settings.
10. Click Add New Hardware, select SCSI Controller, and click OK.
11. Select CD/DVD Drive, and select ISO Image File.
12. Enter the location of the Windows Server 2019 installation media, and click OK.
13. At the prompt, press any key to boot from the CD/DVD location.
14. Click Next.
15. Click Install Now.
16. Select Windows Server 2019 Datacenter Edition, and click Next.
17. Check I accept the license terms, and click Next.
18. Select the VHD, and click Next.
19. After installation, enter a password for Administrator, and click Finish.
20. Shut down the virtual machine.
21. Right-click the virtual machine, and click Delete. Click OK. This does not delete the VHDX file created in step 6.
22. Browse to the location of the VHDX file, and make a copy in the Collect cluster shared volume.

Cloning out the master image

1. Download the VMFleet suite, and extract the contents to a location on the OS drive.
2. From the VMFleet download location, run the following commands to sort the CSVs and install VMFleet:

```
.\install-vmfleet.ps1 -source .
```
3. Copy the 64-bit version of DiskSpd to:

```
C:\ClusterStorage\collect\control\tools
```

 to allow the VMs to download the executable.
4. On one of the cluster nodes, open a PowerShell window as the domain administrator.
5. In PowerShell, navigate to the control folder.
6. Run the following command to clone out 18 VMs per cluster node, or change the value to 24 for 3-node testing.

```
.\create-vmfleet.ps1 -BaseVHD C:\ClusterStorage\collect\VMFleet_gold.vhdx -VMs 18 -Admin Administrator -AdminPass Password1 -ConnectUser Administrator -ConnectPass Password1 -FixedVHD $true -StaticMemory  
get-vm -ComputerName s2dnode1 |set-vm -StaticMemory -ProcessorCount 2 -MemoryStartupBytes 4GB  
get-vm -ComputerName s2dnode2 |set-vm -StaticMemory -ProcessorCount 2 -MemoryStartupBytes 4GB  
get-vm -ComputerName s2dnode3 |set-vm -StaticMemory -ProcessorCount 2 -MemoryStartupBytes 4GB  
get-vm -ComputerName s2dnode4 |set-vm -StaticMemory -ProcessorCount 2 -MemoryStartupBytes 4GB
```

Running VMFleet

We ran the test three times to ensure replicability, and reported the median score.

1. Open a PowerShell window as the domain administrator.
2. In PowerShell, navigate to the control folder.
3. Run the following commands to start the VMs:

```
.\start-vmfleet.ps1
```

4. Begin a test run with the following parameters:

- a. Block size = 8KB
- b. Write ratio = 30%
- c. Random/sequential = Random
- d. Threads = 4
- e. Outstanding IO = 8
- f. Warmup = 10 minutes
- g. Test duration = 20 minutes

```
.\start-sweep.ps1 -b 8 -t 4 -o 8 -w 30 -p r -d 1200 -warm 600
```

Read the report at <http://facts.pt/kjnrftk> ▶

This project was commissioned by HPE.



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.