|  |
| --- |
| Test-DiskSPDBlock Script Guide |
| For Storage Performance Testing  By  Satyajit |

Table of Contents

[Introduction 3](#_Toc436210732)

[What are these Scripts about: 4](#_Toc436210733)

[1. Test-DiskSPDBlock.ps1 4](#_Toc436210734)

[**Here are few defaults that this script considers:** 4](#_Toc436210735)

[**Usage Method:** 5](#_Toc436210736)

[**Sample Run:** 5](#_Toc436210737)

[2. Finetuning-Individual-Storages.ps1 6](#_Toc436210738)

[**Here are few defaults that this script considers:** 6](#_Toc436210739)

[**Usage Method:** 6](#_Toc436210740)

[**Sample Output:** 6](#_Toc436210741)

[3. 64K-Block-Individual-Storages.ps1 7](#_Toc436210742)

[**Sample Logfile:** 7](#_Toc436210743)

[**Here are few defaults that this script considers:** 8](#_Toc436210744)

[**Usage Method:** 8](#_Toc436210745)

[4. 64K-Block-Individual-Storages-Parallel.ps1 8](#_Toc436210746)

[**Here are few defaults that this script considers:** 8](#_Toc436210747)

[**Usage Method:** 8](#_Toc436210748)

[**GroupA.ps1 , GroupB.ps1 , GroupC.ps1 , GroupE.ps1** 9](#_Toc436210749)

[**Nesting is as follows:** 9](#_Toc436210750)

[5. 64K-Block-Individual-Storages-MoveCSV.ps1 10](#_Toc436210751)

[**Test Scenarios:** 10](#_Toc436210752)

[**Here are few defaults that this script considers:** 10](#_Toc436210753)

[**Usage Method:** 10](#_Toc436210754)

[**Nesting is as follows:** 10](#_Toc436210755)

[Improvement Scopes: 10](#_Toc436210756)

[Troubleshooting: 11](#_Toc436210757)

[References: 11](#_Toc436210758)

# Introduction

These scripts are implementation of DiskSPD.exe storage performance testing using PowerShell.

The bundle consists of total of 5 scripts, which serves different testing scenarios and requirements.

Scripts are:

1. Test-DiskSPDBlock.ps1
2. Finetuning-Individual-Storages.ps1
3. 64K-Block-Individual-Storages.ps1
4. 64K-Block-Individual-Storages-Parallel.ps1
   1. GroupA.ps1
   2. GroupB.ps1
   3. GroupC.ps1
   4. GroupE.ps1
5. 64K-Block-Individual-Storages-MoveCSV.ps1

Don’t go by the 64K name, it’s basically named due to the 64k block size value used in the scripts. It can be anything else like 8k, 512k etc.

These scripts are more focused on testing DAS Storage targeted for SharePoint SQL server deployments. But can be used for any other solution, due to the fact being the test doesn’t require you to have any SQL Server pre-installed and it’s directly on Windows Server with storages configured.

In the core it uses DiskSpd.exe a tool from Microsoft for storage testing. The [DiskSPD basics\functionality](http://blogs.technet.com/b/josebda/archive/2014/10/13/diskspd-powershell-and-storage-performance-measuring-iops-throughput-and-latency-for-both-local-disks-and-smb-file-shares.aspx) remains same. The script is more of customization and automation using the common inputs and outputs. Makes using otherwise messy DiskPSD a breeze.

By Default the script expects to find DiskSPD.exe with all its components at this location (Can be modified).

**C:\DiskSPD\amd64fre\diskspd.exe ; C:\DiskSPD**

To get started, you need to download and install the DiskSpd. You can get the tool from <http://aka.ms/DiskSpd> .

You’ll download a ZIP file that contains separate folders for three versions of DiskSpd: amd64fre (for 64-bit systems), x86fre (for 32-bit systems) and armfre (for ARM systems). **To run a test on a server running Windows Server 2012 R2, you only need to download the amd64fre folder.**

* Some of the scripts have ‘Default startup location setting’ and expected script location is this:

**cd C:\DiskSPD\TestScripts**

* The \*Block-Individual-\*.ps1 and Group\*.ps1 scripts have usage of #Get-ClusterSharedVolume, should be commented # if Failover Cluster is not in use.

IMPORTANT: - The PowerShell Session should be ‘Run as administrator’. Otherwise you would see strange errors and crashes.

# What are these Scripts about:

## Test-DiskSPDBlock.ps1

This is the main script and heart of the bundle, does the parameter setting on DiskSPD.exe and formats the output. Already has default values set, hence can be run without any parameters if required.

Feature Highlights:

1. Read, Write, RW testing.
2. Combined (Shared) or Individual Paths testing.
3. Multiple IODs set or none declared.
4. Various switches and parameters as per usages (No editing on the script).
5. Defaults values on common parameters.
6. Execute without any parameters (creates testfile.dat in current directory).

To know more about the script usage, run the below command in PowerShell:

help .\Test-DiskSPDBlock.ps1 -Full

There is extensive help built into the script.

PS C:\DiskSPD\TestScript> help .\Test-DiskSPDBlock.ps1 -Full

NAME

C:\Script\Test-DiskSPD\Test-DiskSPDBlock.ps1

SYNOPSIS

This script uses DiskSPD.exe to test the storage and formats the output.

Already has default values hence can be run without any parameters.

SYNTAX

C:\Users\satyajit.baruah\Desktop\Sdata\Script\Test-DiskSPD\Test-DiskSPDBlock.ps1 [[-Truns] <Int32>] [[-Time] <Object>] [[-Threads] <Object>] [[-Size] <Object>]

[[-BlockSize] <Object>] [[-Paths] <Object>] [[-IODs] <String[]>] [[-RWPercent] <Object>] [-SeqIO] [-ReadTest] [-WriteTest] [-RWTest] [-FineTuning] [-SharedTest]

[<CommonParameters>]

Approx. Run Time: 10 mins for each path\ testfile.dat

### **Here are few defaults that this script considers:**

Recommend not to change it unless a parameter is not available to set it. All these can be overridden via parameter and switches.

* Total Runs\Tests Loops: 10 (useful for getting average values)
* Test File size(-c): 100 GB
* Time (-d): 60 sec
* Threads(-t): 4
* BlockSize(-b): 512k
* Path: "testfile.dat" (In current folder .\)
* $IODs(-o): 14
* By default only R/W Percent(-w): 50% / 50% is done
* Random I/O (-r) for each test
* Individual serial tests if multiple targets (-SharedTest: $false), unlike DiskSPD.
* -h Caching disabled, -L Measure latency statistics enabled

### **Usage Method:**

Open PowerShell as Administrator or call it as a nested script:

.\Test-DiskSPDBlock.ps1

### **Sample Run:**

$iods="4"

$paths="C:\ClusterStorage\DiskGroupE\testfile.dat"

.\Test-DiskSPDBlock.ps1 -BlockSize 64K -Paths $paths -IODs $iods -Time 60 -Truns 10 -RWPercent 30

Starting Large-512k-SERVER-SQL-1 test on Thursday, November 19, 2015 12:37:13 AM

50% Read/50% Write Test on testfile.dat at 12:37 AM

Variables: -c100G -d1 -r -w50 -b512k -o14 -t4 -h -L testfile.dat

Run IOPS Throughput(MB/sec) Latency(ms) CPU(%)

--- ----- ------------------ ----------- ------

1 9.98 4.99 4057.809 1.56

2 12.99 6.50 3232.656 3.12

--- -------- ------ ------- -----

Average 11.485 5.745 3645.2325 2.34

Finished Large-512k-SERVER-SQL-1 test on Thursday, November 19, 2015 12:37:33 AM

Read\Write Test Average:

Path IOPS Throughput(MB/sec) Latency(ms) CPU(%)

---- ---- ------------------ ----------- ------

1 11.485 5.745 3645.2325 2.34

Test duration: 0days 0hrs 0mins and 20secs

## Finetuning-Individual-Storages.ps1

This script is not detailed, customized or parameterized as the Test-DiskSpd script, but instead focuses on the formatting and suggesting the top 3 IOD values to be used.

The suggestion is based on Latency < 20 ms and\or CPU < 80%

**NOTE:-**This script doesn't create the testfiles.dat, need to run other script first with any IOD values. .\Test-DiskSPDBlock.ps1 can be run without any parameters to create the files. It uses default IOD of 14.

Customization needs to be directly done on this section and some variables:

$result = C:\DiskSPD\amd64fre\diskspd.exe $time -r -w30 -t4 $param -b64K -h -L $path

$paths variable needs to be set with space separated full paths, just like Diskspd.exe.

Also it doesn’t have option to test a combined multiple paths SharedTest. Needs a small tweak by removing the $paths for-loop. Replacing it with $path = $paths, similar to the Test-DiskSPD.ps1

Approx. Run Time: 60 mins for each path\ testfile.dat

### **Here are few defaults that this script considers:**

Time(-d): 15 secs , 70%read/30%write, Random -r , IOD Range tested [1.. 45] , Threads(-t): 4, BlockSize(-b): 64K

### **Usage Method:**

1. Set the $paths variable with the new path(s).
2. Open PowerShell as Administrator:

.\Finetuning-Individual-Storages.ps1

### **Sample Output:**

Suggested Group Only Latency < 20 limit:

IOD IOPS Throughput\_MBps Latency\_ms CPU\_%

--- ---- --------------- ---------- -----

22 18531.58 1158.22 4.747 20.29

38 18524.50 1157.78 8.196 14.04

18 18513.06 1157.07 3.886 12.49

Suggested Group Latency < 20 and CPU < 30 limit:

IOD IOPS Throughput\_MBps Latency\_ms CPU\_%

--- ---- --------------- ---------- -----

22 18531.58 1158.22 4.747 20.29

38 18524.50 1157.78 8.196 14.04

18 18513.06 1157.07 3.886 12.49

Small 64K, 4 thread ending

Thursday, November 19, 2015 12:00:19 AM

Test duration

Days Hours Minutes Seconds

---- ----- ------- -------

0 0 18 27

## 64K-Block-Individual-Storages.ps1

This script is basically collating all the inputs together and passing it over to Test-DiskSPD for testing and generating results. It’s also responsible for saving the results to the Log file using Start-Transcript.

Here we would use the best IOD value for individual paths from the finetuning script and the full path names in $iods and $paths variables to be passed on to Test-DiskSPD.ps1

Logs files are automatically named as per the calling script and stored in **<currentpath>\Logs\** folder by default.

### **Sample Logfile:**

64K-Block-Individual-Storages.ps1\_2015-10-14\_09.04.23AM.txt

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Windows PowerShell transcript start

Start time: 20151014090423

Username: UCC\tst\_user

RunAs User: UCC\tst\_user

Machine: SERVER-SQL-1 (Microsoft Windows NT 6.2.9200.0)

Host Application: C:\Windows\System32\WindowsPowerShell\v1.0\powershell\_ise.exe .\64K-Block-Individual-Storages.ps1

Process ID: 2040

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Transcript started, output file is .\Logs\64K-Block-Individual-Storages.ps1\_2015-10-14\_09.04.23AM.txt

Starting Small-64K-SERVER-SQL-1 test on Wednesday, October 14, 2015 09:04:23 AM

This script is also used as a base script for some other scripts as well.

Approx. Run Time: 10 mins for each path\ testfile.dat

### **Here are few defaults that this script considers:**

70% Read / 30% Write Individual Test, Random default, Thread 4 default, BlockSize 64k, Time 60 sec.

Remaining [defaults](#_Here_are_few) are taken from the main Test-DiskSPD.ps1 script.

### **Usage Method:**

1. Update the $iods and $paths with new values
2. Open PowerShell as Administrator:

PS C:\DiskSPD\TestScripts> .\64K-Block-Individual-Storages.ps1

## 64K-Block-Individual-Storages-Parallel.ps1

This script allows to make parallel calls on the individual Group A, B, C and E scripts in new sessions\windows. This helps in quickly start the test with a single click.

This is different than the DiskSPD multi-threading parallel execution which is limited to single \ common storage.

Here each group can execute on their respective storages in parallel. There are literally 4 DiskSPD.exe running at the same time.

The PSsession title bar shows the ‘Group Names’ for easy reference. The script is not set to close the window after execution is complete. Every execution starts fresh window and data is logged as usual for viewing later.

The main scripts waits until you close the output windows.

Approx. Run Time: 10 mins

### **Here are few defaults that this script considers:**

* Default startup location cd C:\DiskSPD\TestScripts
* Expects the Group\*.ps1 scripts in the current location

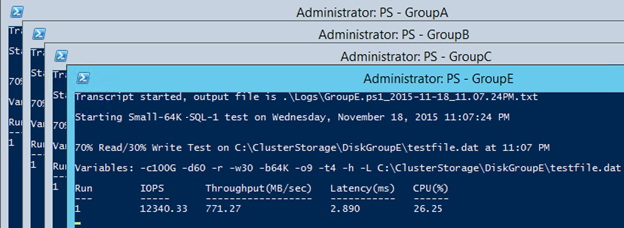
### **Usage Method:**

Open PowerShell as Administrator:

PS C:\DiskSPD\TestScripts> .\64K-Block-Individual-Storages-Parallel.ps1

### **GroupA.ps1 , GroupB.ps1 , GroupC.ps1 , GroupE.ps1**

These scripts are nothing but copy of the same ‘64K-Block-Individual-Storages.ps1’ script. Only difference being individual\single path used, instead of all paths looping sequentially from one script.



#### **Here are few defaults that these script considers:**

70% Read / 30% Write Individual Test, Random default, Thread 4 default, BlockSize 64k, Time 60 sec. Remaining [defaults](#_Here_are_few) are taken from the main Test-DiskSPD.ps1 script.

#### **Usage Method:**

1. Update the $iods and $paths with new values (single path expected in each file called)
2. Open PowerShell as Administrator or directly called inside another script:

.\GroupA.ps1

At the moment no need to run this script separately.

### **Nesting is as follows:**

64K-Block-Individual-Storages-Parallel.ps1 -> GroupA,B,C,E .ps1-> Test-DiskSPDBlock.ps1

## 64K-Block-Individual-Storages-MoveCSV.ps1

This script moves the CSVs automatically and test all the 4 scenarios. Behind the scenes it wraps ‘64K-Block-Individual-Storages.ps1’ script to do the actual testing.

### **Test Scenarios:**

#1 –Split VDs - 50\50 split - 1,2,1,2

#2 – Switch the VDs to the opposing HOST, SQL (split VDs)

#3 – All VDs assigned to SERVER-HOST-1, SERVER-SQL-1

#4 – All VDs assigned to SERVER-HOST-2, SERVER-SQL-2

Approx. Run Time: 3hrs

The goal is to get Individual storage tested across all combinations of HOST and VM activation scenarios from a single VM node.

This test should be run from VM-1, then again from VM-2 if required to compare the results.

### **Here are few defaults that this script considers:**

Default startup location: cd C:\DiskSPD\TestScripts. Remaining defaults are from the nested script.

### **Usage Method:**

1. Update the cluster names, server node, #comment lines if nested VM cluster(SPDAS) is not used.
2. Update the $iods and $paths with new values in ‘64K-Block-Individual-Storages.ps1’
3. Open PowerShell as Administrator:

PS C:\DiskSPD\TestScripts> .\64K-Block-Individual-Storages-MoveCSV.ps1

### **Nesting is as follows:**

64K-Block-Individual-Storages-MoveCSV.ps1 -> 64K-Block-Individual-Storages.ps1 -> Test-DiskSPDBlock.ps1

# Improvement Scopes:

* The logging \ saving the contents to file method is quite raw using Start-Transcript can be improved using Out-File or something more controlled.
* Get Output as custom PSObjects, instead specialized formatted text
* Re-enable the Progress Bar, code is already present, but nesting scripts makes it not much useful. Hence lies currently disabled.

# Troubleshooting:

* **Starting the script crashes the Diskspd.exe:**

Either you are not running on admin mode, or the testfile has already been locked by some other running Diskspd test. You can’t run test on the same Testfile.dat concurrently.

* **Resulting log files are skewed with garbage data and getting overwritten randomly:**

Apparently transcripts keep running overwriting the data, if script is not stopped properly on the PS Session.

Either let the script finish, if Ctrl+C is used to break the execution you need to stop the transcripts or start a fresh PSSession.

As a pre-cautionary measure, before starting a script. Stop-Transcript should be run until we get this error.

*Stop-Transcript : An error occurred stopping transcription: The host is not currently transcribing.*

# References:

* [DiskSpd, PowerShell and storage performance](http://blogs.technet.com/b/josebda/archive/2014/10/13/diskspd-powershell-and-storage-performance-measuring-iops-throughput-and-latency-for-both-local-disks-and-smb-file-shares.aspx)
* [Diskspd, a Robust Storage Testing Tool, Now Publically Available Download](http://aka.ms/DiskSpd)

(<http://aka.ms/DiskSpd> )