

[Powershell Scripts for EMC Performance Statistics](#)

Clint Kitson - EMC vSpecialist - @clintonskitson

EMC Community Network - Everything VMware at EMC

https://community.emc.com/community/connect/everything_vmware?view=overview

02/27/11 Version 1.0.1 (Limited testing so far, provided as-is and suggested currently for non-production and lab use only, see MD5 checksums for the zip files at above URL)

TABLE OF CONTENTS

- Overview
- Powershell Scripts for Performance Statistics
 - Celerra
 - Clariion
 - Symmetrix
- Outputting to a CSV file
- Leveraging Powershell and Collected Statistics

Overview

Introduction

This document is written to discuss and provide information around running powershell scripts for collecting and viewing performance statistics on EMC storage arrays. Each script has default runtime behavior that will return more generic results from each array. As well there are inputs that can be sent to the script which will allow for the filtering of array objects. For example, if you have a NAS export name, UUID, or name of a device the script will filter for relevant objects and return just those items. If these items aren't specified then most other items will be returned.

The scripts are written completely in powershell and leverage native commands from EMC management utilities per array type to obtain the proper information. This means there is a dependency for each script that the management software from the array you are trying to pull statistics from is running and available and powershell is installed and properly allowing execution of scripts. See the details below per script. The scripts themselves are not recording any statistics information. However, there is a section below that gives an example of what extra commands to add at the end of powershell execution to output to files and "other things."

Performance stats are derived (script calculated) or calculated. The sampling time starts when the script executes, so the minimum time to complete the script is based on the sampling interval.

Runtime Abilities

The scripts will by default output information in a standardized manner as a powershell object. This allows for a lot of options for formatting and using the output in other ways. The runtime options for each script can be simple-- which array to target and how to authenticate. From there you have optional parameters to specify intervals, sleep times, and specific objects to query.

Security

I highly suggest that you create non-privileged monitoring accounts to access the management software on each array. Symmetrix and Celerra monitoring utilizes SSH to pull information. The Clariion leverages a local install of naviseccli and can use security files instead of specifying authentication. However, commands are still placed in open scripts so it is possible to inject malicious code into command statements. Please consider this and checking MD5 signatures of zip files before using.

Runtime Performance Notes

There is a high amount of processing that takes place on the powershell machine and some parallelization that occurs when kicking off collections simultaneously. Make sure the system running the powershell commands has available memory and CPU processing to ensure proper operations. On the target side there are multiple collection commands executed at the same time. This can cause some load on the target device, so it is important to ensure that the target array responds appropriately to information gathering requests. Also make sure for the Symmetrix and Celerra runtime that your reverse lookups are either working correctly or disabled via the SSH service. If not, there will be a 3-5 second delay for SSH sessions due to reverse lookups timing out.

Runtime options for all scripts

-username

- objects = server_stats -l lists all of the available objects that can be monitored
- nfsexport = The export name that is the focus of monitoring. With this command, there will be specific volumes related to the export showed.
- datamover (default server_2) = Specify the datamover to query, if you specify nfsexport then this will be ignored and determined automatically
- sleep (5 seconds default) = amount of time to wait in between set of performance samplings
- objects = Which performance objects should be queried, "server_stats server_2 -" will list available queries. This is not the same as -metrics which is purely a choice of which columns or metrics to be shown.

Runtime Examples

(.\get_unified_nas_perf.ps1 -csip csip -username xx -password xx) – Return default performance statistics for any object that has metrics for that time period.

(.\get_unified_nas_perf.ps1 -csip csip -username xx -password xx -objects net.device,store,sys.cpuUtil,nfs.basic) – Return these specific performance stats (dart5 can be different than dart6 on objects, for example sys.cpuUtil is different per)

(.\get_unified_nas_perf.ps1 -csip csip -username xx -password xx -nfsexport exportname -objects net.device,store) – Return specific stats for a datamover and dependent volumes around a specific NFS export.

Preflight Checks

SSH to the Control Station as the correct user. Run (export NAS_DB=/nas) if not logging in as nasadmin. Run (/nas/bin/server_stats server_2). Live statistics should return.

Available Metrics

Name,Util %,Network In KiB/s,Network Out KiB/s,Network In Pkts/s,Network Out Pkts/s,Network In Errors/s,Network Out Errors/s,Queue Depth,Read Ops/s,Write Ops/s,Read KiB/s,Write KiB/s,Avg Read Size Bytes,Avg Write Size Bytes

Others stats are available, but the script is made mainly for these right now.

EMC Clariion (BLOCK) Statistics – get_unified_block_perf

Name	Prct Busy	Reads	Writes	Read MB	Write MB	IO Size	Read KB	IO Size	Write KB	Read Cache Hits	Write Cache Hits	Prefetched B	Forced Flushes	Write Cache Flashes MB	FRST Cache Read Hits	FRST Cache Write Hits	FRST Cache M
SP 0	10.64	6.37	39.85	0.148	0.405	21.728	513.073		10.401					0.01	8.14	2.08	
LUN 1	0.63	1.16		0.581													
LUN 2004																	
LUN 2006																	
LUN 2010																	
LUN 2014																	
LUN 2016																	
LUN 2018																	
nv_demo_01																	
TEST01																	
TEST02																	
TEST03																	
TEST04																	
TEST05																	
TEST06																	
TEST07																	
TEST08																	
TEST09																	
TEST10																	
WC-V1-ds04																	
WC-V1-poo...																	
WC-V1-poo...																	
SP 0	8.57	0.83	5.4	0.022	0.028	27.211	513.073		5.29								
LUN 0	0.63	1.16		0.581													
LUN 0001																	
LUN 0002																	
LUN 0003																	
LUN 0004																	
LUN 0005																	
LUN 0006																	
LUN 0007																	
LUN 0008																	
LUN 0009																	
LUN 0010																	
LUN 0011																	
LUN 0012																	
LUN 0013																	
LUN 0014																	
LUN 0015																	
WC-V1-poo...																	

PS D:\projects>ps_get_navisec_perf\012511> .\get_naviseccli_perf.ps1 -user admin -password [REDACTED] -scope 0 -spaip 10.5.200.241 -sobjip 10.5.200.242

Introduction

The `get_unified_block_perf` script will return block performance statistics from a NS/CX/Clariion array around SP and LUN objects with metrics around utilization, IOs, bandwidth, cache, and FASTCache stats. Traditional flare LUNs will return extensive statistics. Thin/thick luns will return statistics around the virtual LUNs from the raidgroups of storage pools and not directly the thin/thick luns themselves. These stats require an option in the runtime for engineering mode.

Dependencies

The script requires that Navisphere Server Software for Windows is installed. The script was tested against “**Windows Navisphere Server Software 7.30**”. You can download from Powerlink -> Support -> Software Downloads and Licensing -> Download J-O -> Navisphere Server Software. Copy the `navisecli.exe` executable after install from the Program Files (x86)-> EMC -> Navisphere CLI directory to the directory where the scripts are being run from. In order for the array to be collecting statistics, Statistics Gathering and Data logging must be enabled and started. Statistics can be enabled from Unisphere -> Properties -> Statistics checkbox and Data logging can be enabled from Monitoring -> Analyzer -> Performance Data Logging -> Start. Analyzer does not need to be licensed to use this script.

Runtime

-spaip = Storage Processor A management IP
-spbip = Storage Processor B management IP
-scope # = # is normally 0 for global, see `navisecli` documentation

(optional)

-usesecfile = Use pre-determined authentication files. This can be run with “`navisecli -user admin -password password -scope 0 -h spaip -addusersecurity`”. Make sure to this twice, once for SPA and once for SPB. With this specified -username and -password and -scope are not needed.
-eng = This switch must be followed by the engineering password for the Clariion. It will return more metrics and objects (virtual luns from pools).
-lunid – Return only specific information based on this LUN

Runtime Examples

`(.\get_navisecli_perf.ps1 -spaip spaip -spbip spbip -username admin -password pass -scope 0)`

Preflight Checks

After installing Navisphere Server Utility as described above and copied the `navisecli.exe` file to the script directory run (`navisecli -user xx -password xx -h spip -scope 0 getall -sp`). It should return information for that SP. Run the command on the second SP as well and make sure they both are responding as expected. It is also a good idea if you are using a security file to test the above command as (`navisecli -h spip getall -sp`) for both SPs.

Available Metrics

EMC Symmetrix (BLOCK) Statistics - get_symcli_perf

Name	r_per_second	w_per_second	io_per_second	kb_r_per_second	kb_w_per_second	r_cache_hit_pct	w_cache_hit_pct	rw_cache_hit_pct	sequential_r_pct	wb_tracks
Director: DF-T-1		3								
Director: DF-T-2										
Director: DF-T-3										
Director: DF-T-4										
Director: DF-T-5	2	4	5					100		
Device: 990										
Device: 991					4			100		
Device: 992					24			100		
Device: 993				64	18			100		
Device: 994				1	34			100		
Device: 995					15			100		
Device: 996					34			100		
Device: 997					10			100		
Device: 998		11		64	7			100		
Device: 999		12			18			100		
Device: 9900	1	10		81	25	100		100		
Device: 9901					19			100		
Device: 9902					8			100		
Device: 9903	1	11			11			100		
Device: 9904					34			100		
Disk: 07C-D-0	9	8			173			100		
Disk: 07C-D-1					6			100		
Disk: 07C-D-2		2			136			100		
Disk: 07C-D-3					136			100		
Disk: 07C-D-4	1				34			100		
Disk: 07C-D-5					115			100		
Disk: 07C-D-6	8	8			136			100		
Disk: 07C-D-7					2			100		
Disk: 07C-D-8		19			1245			100		
Disk: 07C-D-9					2			100		
Disk: 07C-D-10					512			100		
Disk: 07C-D-11					2			100		
Disk: 07C-D-12					136			100		
Disk: 07C-D-13					2			100		
Disk: 07C-D-14		2			136			100		
Disk: 07C-D-15					34			100		
Disk: 07C-D-16	7	7			115			100		
Disk: 07C-D-17					137			100		
Disk: 07C-D-18					34			100		
Disk: 07C-D-19	7	7			115			100		
Disk: 07C-D-20					141			100		
Disk: 07C-D-21		2			136			100		

Introduction

The get_symcli_perf command will return performance statistics from a Symmetrix array around director, device, and disk objects with metrics around IO, bandwidth, and cache.

Device and grab metas and disks

Dependencies

The script executes commands via SSH to a system with symcli Solutions Enabler Linux installed and working to the target array. The easiest method is to download the vApp and follow along with the documentation. <http://powerlink.emc.com> -> Support -> Software Downloads and Licensing -> Downloads S -> Solutions Enabler -> se___vapp.zip

Runtime

-symclihost – Target symcli host

(optional)

-lunid – Return only specific information based on this LUNID

-dev_name – return only specific devices making up the meta based on this primary device name

-sleep (5 seconds default) = amount of time to wait in between set of performance samplings

Runtime Examples

(.\get_symcli_perf.ps1 -symclihost symclihost -username user -password pass –paint) – Returns all statistics that the stats command collects data for in the default 15 second interval and paints them on the screen in a continuous fashion.

Preflight Checks

After getting the solutions enabler and symcli working, SSH to the system that is running the symcli software. From there execute (symstat -type REQUEST -i 5 -c 1). You should see a table with some columns returned and if there is activity, stats returned as well.

Available Metrics

Name, r_per_second, w_per_second, io_per_second, kb_r_per_second, kb_w_per_second, r_cache_hit_pct, w_cache_hit_pct, rw_cache_hit_pct, sequential_r_pct, wp_tracks

Outputting to a CSV file

Each script has the parameter of -outcsv where you can enter a csv file to output the performance samplings to. These stats can then be compressed and sent via email, or you can leverage another script to monitor the CSV files. More to come below on that..

Leveraging Powershell and Collected Statistics

Next