Instructions for Virtual Machine Compute Optimizer

Last revised: 5/7/2020

VMCO Version: 2.1.0

Contents

Summary

Understanding Virtual Machine vCPU and vNUMA Rightsizing and Host Power Management Policies for Best Performance

Virtual Machine vNUMA configuration and Host Management Power Policies can have a dramatic effect on individual and overall VM performance on a host. The effects of a physical host's NUMA architecture on virtual machines' vNUMA configuration has been covered in-depth in Mark Achtemichuk's Blog Virtual Machine vCPU and vNUMA Rightsizing – Rules of Thumb. The Host Power Management in ESXi is covered in Performance Best Practices for VMware vSphere 6.7. These are applicable to any supported version of ESXi currently available.

What's New in Version 2.0.4?

- Fixed errors in reporting for some VMs that are on hosts with 4 sockets
- Fixed "memory" missing from Details when VM memory spans pNUMA nodes
- Added ability to call function with "-simple" which only reports VM info (leaves out vCenter, Cluster, and Host)

What Is the Virtual Machine Compute Optimizer?

The VMCO is a Powershell script that uses the PowerCLI module to capture information about the VMs, Hosts, and clusters running in your vSphere environment, and reports back on whether the VMs are configured optimally based on the VM configuration, Host configuration, and Cluster Host minimums. It will flag a VM as "YES" if it is optimized and "NO" if it is not. For non-optimized VMs, a recommendation is made that will keep the same number of vCPUs currently configured, with the optimal number of virtual cores and sockets. In some cases, the vCPUs could be configured correctly, but other VM, Host, or Cluster configuration that could impact performance will be called out.

Note that the VMCO will not analyze whether your VMs are configured with the correct number of vCPUs based on the VM's workload. A more in-depth analysis tool such as VMware vRealize Operations Manager can make right-sizing determinations based on workload and actual performance.

Components

The VMCO consists of two Powershell scripts.

1. Virtual_Machine_Compute_Optimizer_vX.Y.Z.ps1
    a. Use this if you want to analyze entire vCenters as it automates the process to
        i. connect to your vCenters & set multi-vcenter mode
        ii. verify your Powershell version
        iii. verify PowerCLI Module is installed and update it, or it will download/install it for you
        iv. checks for, automatically loads, and calls the Get-OptimalvCPU function and exports the results to a CSV file that you specify
2. Get-OptimalvCPU_vX.Y.Z.ps1
    a. Use this is you want to

i. only analyze specific VMs
ii. choose what to do with the results
iii. include the function in your own script
iv. use -simple option to only show VM info

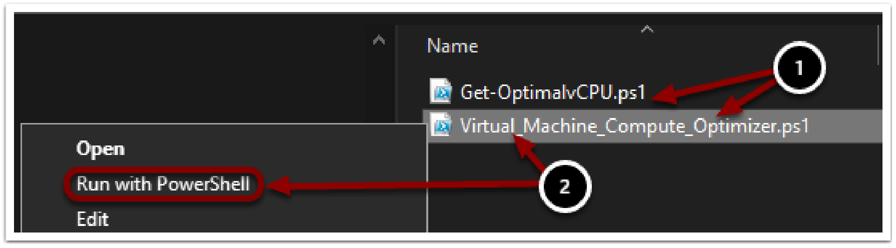
Running the Virtual Machine Compute Optimizer

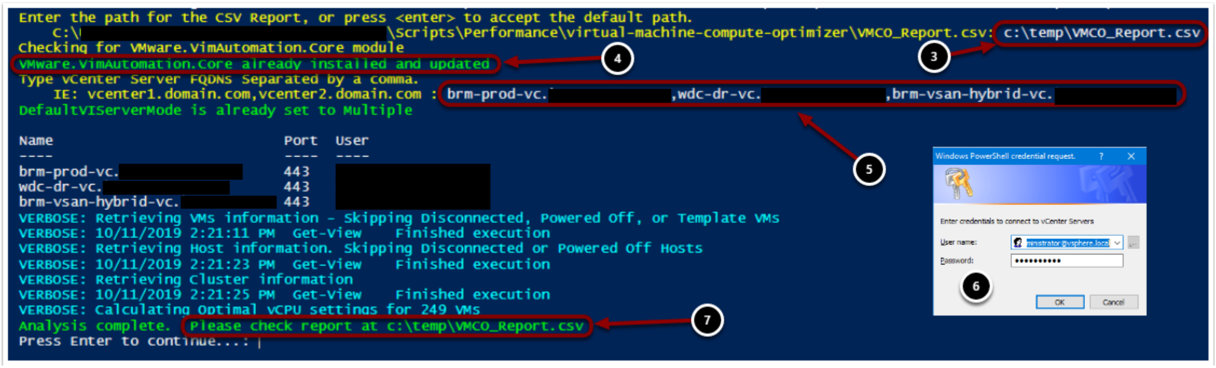
Requirements

In order to run the Virtual Machine Compute Optimizer, you will need the following:

- Windows 7/Server 2008 or above
- Powershell v5 or higher
- The PowerCLI Module installed, or access to the internet. The VMCO will attempt to install the module if it is not already, and give you an option to upgrade if it is out of date.
- A user account with Read-Only rights assigned at the vCenter level with 'Propagate to children' enabled. These rights will be needed on each vCenter that will be analyzed.
- The two Powershell scripts "Virtual_Machine_Compute_Optimizer_vX.Y.Z.ps1.ps1 " and "Get-OptimalvCPU_vX.Y.Z.ps1.ps1" in the same folder.

Steps





```
Enter the path for the CSV Report, or press <enter> to accept the default path.
    C:\[redacted]\Scripts\Performance\virtual-machine-compute-optimizer\VMCO_Report.csv: c:\temp\VMCO_Report.csv
Checking for VMware.VimAutomation.Core module
VMware.VimAutomation.Core already installed and updated
Type vCenter Server FQDNs separated by a comma.
    IE: vcenter1.domain.com,vcenter2.domain.com : brm-prod-vc.[redacted],wdc-dr-vc.[redacted],brm-vsan-hybrid-vc.[redacted]
DefaultVIServerMode is already set to Multiple

Name                        Port  User
----                        ----  ----
brm-prod-vc.[redacted]      443   [redacted]
wdc-dr-vc.[redacted]        443
brm-vsan-hybrid-vc.[redacted] 443
VERBOSE: Retrieving VMs information - Skipping Disconnected, Powered Off, or Template VMs
VERBOSE: 10/11/2019 2:21:11 PM  Get-View    Finished execution
VERBOSE: Retrieving Host information. Skipping Disconnected or Powered Off Hosts
VERBOSE: 10/11/2019 2:21:23 PM  Get-View    Finished execution
VERBOSE: Retrieving Cluster information
VERBOSE: 10/11/2019 2:21:25 PM  Get-View    Finished execution
VERBOSE: Calculating Optimal vCPU settings for 249 VMs
Analysis complete.  Please check report at c:\temp\VMCO_Report.csv
Press Enter to continue...: |
```

Windows PowerShell credential request.
Enter credentials to connect to vCenter Servers
User name: ministrator@vsphere.local
Password: ••••••••••
OK
Cancel
3
4
5
6
7

There are a few ways you can launch a Powershell script. We will use the simplest approach.

1. Download the "Virtual_Machine_Compute_Optimizer_vX.Y.Z.ps1" and the "Get-OptimalvCPU_vX.Y.Z.ps1.ps1" files and put them in the same directory.
2. Right click the "Virtual_Machine_Compute_Optimizer_vX.Y.Z.ps1" file and choose "Run with Powershell"
3. When prompted, type in the full path to a csv file where you would like to store the report results. By hitting enter, the path will default to the location you are running the script from,

with a filename of “vNUMA_Report.csv”. ie: “<Path_To_Script>\vNUMA_Report.csv”. If the file already exists, it will attempt to merge the new report data into it. This could be helpful to combine other vCenter reports that might have been run separately.

4. The Core PowerCLI Module is needed to run this script. If it is not installed, you will be prompted to install it. If it is not the current version, you will be prompted to update. Although you don’t have to choose to update the module, it does have to be installed. You will see 1 of 3 outputs on the screen:
    a. VMware.VimAutomation.Core already installed and updated
    b. VMware.VimAutomation.Core is not installed. Type 'Y' to install module, or 'N' to skip:
    c. A newer version of VMware.VimAutomation.Core is available. Type 'Y' to update module to x.y.z or 'N' to continue with current version a.b.c:

       *the computer you are running the script from has to have internet access in order to install or upgrade the module. You will have to manually install the module otherwise.

5. When prompted, enter the FQDN or the vCenter(s) you would like to analyze. Separate the names with a comma “,”. IE, vcenter1.domain.com,vcenter2.domain.com,vcenter3.domain.com
6. You will be prompted to enter credentials to connect to the vCenter(s). The user account must have the rights defined under the Requirements section. You cannot specify different credentials per vCenter at this time.
7. Once authenticated, the progress will show on the screen. When the analysis is complete, it will indicate the location of the report file.

Using the Get-OptimalvCPU Function

Requirements

In order to run the Get-OptimalvCPU Function, you will need the following:

- Windows 7/Server 2008 or above
- Powershell v5 or higher
- The PowerCLI Module installed
- A user account with Read-Only rights assigned at the vCenter level with ‘Propagate to children’ enabled. These rights will be needed on each vCenter that will be analyzed.
- The “Get-OptimalvCPU_vX.Y.Z.ps1” file

Examples

For using the function, there is some assumption that you understand how to load a Powershell function, and know how to connect to the vCenter servers you want to query via the PowerCLI Core Module. Below is the output from calling “get-help Get-OptimalvCPU -examples” command.

Note that if you run “Get-OptimalvCPU with no parameters, it will retrieve all VMs from currently connected vCenters.

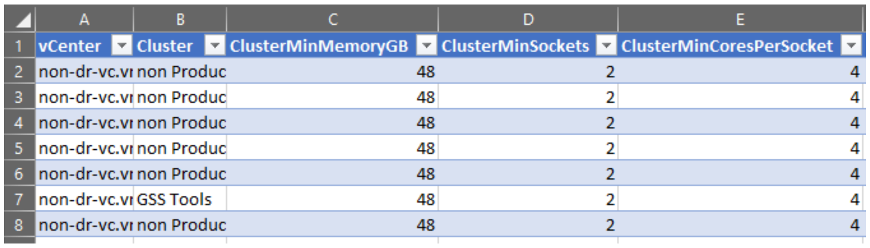
```
PS C:\> get-help Get-OptimalvCPU -examples

NAME
    Get-OptimalvCPU

SYNOPSIS
```

```
    Calculates the optimal vCPU (sockets & cores) based on the current VM and Host
architecture


    -------------------------- EXAMPLE 1 --------------------------

    PS C:\>Get-OptimalvCPU      #Gets all VMs from currently connected vCenters

    -------------------------- EXAMPLE 2 --------------------------

    PS C:\>Get-OptimalvCPU | Export-CSV -path "c:\temp\vNUMA.csv" -NoTypeInformation


    -------------------------- EXAMPLE 3 --------------------------

    PS C:\>Get-OptimalvCPU -VMName "MyVmName"

    -------------------------- EXAMPLE 4 --------------------------

    PS C:\>Get-OptimalvCPU -VMName (Get-VM -Name "*NY-DC*" | Select -expand Name)
```

Interpreting the Results

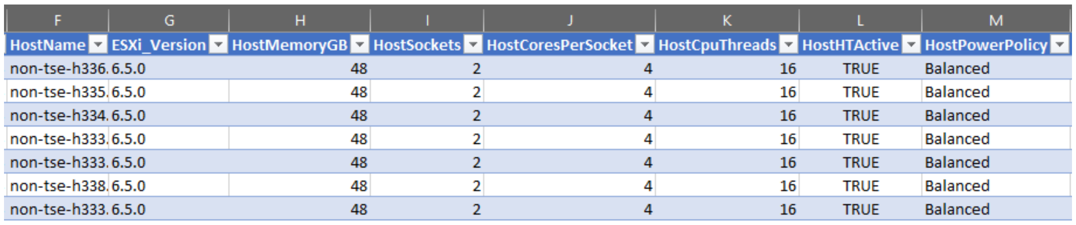
The report that is generated is a simple CSV file. It includes the data collected from the vCenters, Clusters, Hosts, and VMs. Let's break down the data.

Columns A through E contain the vCenter Name, the Cluster Name, and the Cluster's minimum Host Memory, CPU Sockets, and CPU Cores. The minimums are used to calculate if the cluster contains inconsistent Hosts in the cluster, which can affect how vNUMA is presented to the guest.



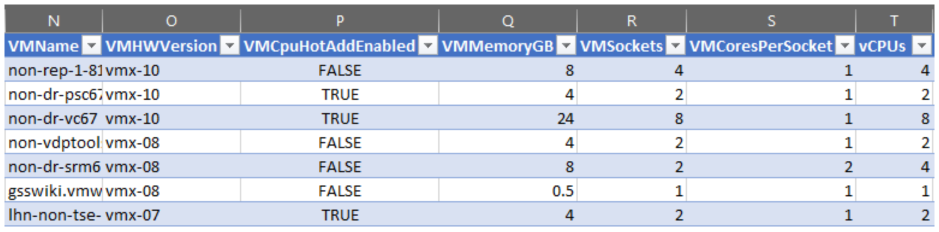
	A	B	C	D	E
1	vCenter	Cluster	ClusterMinMemoryGB	ClusterMinSockets	ClusterMinCoresPerSocket
2	non-dr-vc.v	non Produc	48	2	4
3	non-dr-vc.v	non Produc	48	2	4
4	non-dr-vc.v	non Produc	48	2	4
5	non-dr-vc.v	non Produc	48	2	4
6	non-dr-vc.v	non Produc	48	2	4
7	non-dr-vc.v	GSS Tools	48	2	4
8	non-dr-vc.v	non Produc	48	2	4

Columns F through M contain the Host information collected that are used in calculations.



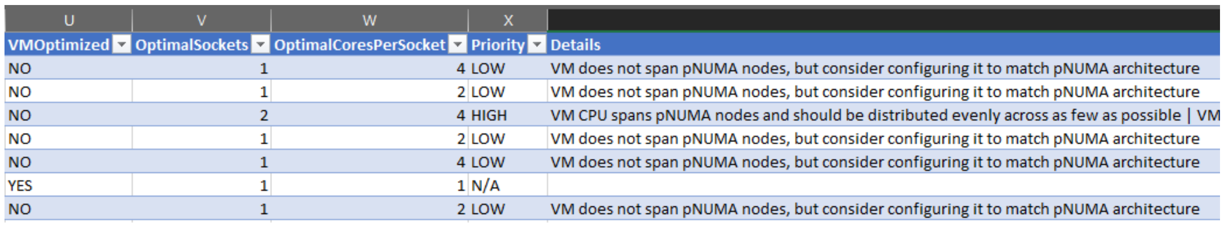
F	G	H	I	J	K	L	M
HostName	ESXi_Version	HostMemoryGB	HostSockets	HostCoresPerSocket	HostCpuThreads	HostHTActive	HostPowerPolicy
non-tse-h336.	6.5.0	48	2	4	16	TRUE	Balanced
non-tse-h335.	6.5.0	48	2	4	16	TRUE	Balanced
non-tse-h334.	6.5.0	48	2	4	16	TRUE	Balanced
non-tse-h333.	6.5.0	48	2	4	16	TRUE	Balanced
non-tse-h333.	6.5.0	48	2	4	16	TRUE	Balanced
non-tse-h338.	6.5.0	48	2	4	16	TRUE	Balanced
non-tse-h333.	6.5.0	48	2	4	16	TRUE	Balanced

Columns N through T contain the VM information collected that are used in calculations



N	O	P	Q	R	S	T
VMName	VMHWVersion	VMCpuHotAddEnabled	VMMemoryGB	VMSockets	VMCoresPerSocket	vCPUs
non-rep-1-81	vmx-10	FALSE	8	4	1	4
non-dr-psc67	vmx-10	TRUE	4	2	1	2
non-dr-vc67	vmx-10	TRUE	24	8	1	8
non-vdptool	vmx-08	FALSE	4	2	1	2
non-dr-srm6	vmx-08	FALSE	8	2	2	4
gsswiki.vmw	vmx-08	FALSE	0.5	1	1	1
lhn-non-tse-	vmx-07	TRUE	4	2	1	2

Finally, Columns U through Y contain the recommendations, priority, and details of the optimal vCPU calculations.



U	V	W	X	
VMOptimized	OptimalSockets	OptimalCoresPerSocket	Priority	Details
NO	1	4	LOW	VM does not span pNUMA nodes, but consider configuring it to match pNUMA architecture
NO	1	2	LOW	VM does not span pNUMA nodes, but consider configuring it to match pNUMA architecture
NO	2	4	HIGH	VM CPU spans pNUMA nodes and should be distributed evenly across as few as possible | VM
NO	1	2	LOW	VM does not span pNUMA nodes, but consider configuring it to match pNUMA architecture
NO	1	4	LOW	VM does not span pNUMA nodes, but consider configuring it to match pNUMA architecture
YES	1	1	N/A	
NO	1	2	LOW	VM does not span pNUMA nodes, but consider configuring it to match pNUMA architecture

VMOptimized

- Yes – The VM is configured optimally, and no changes need to be made
- No – The VM needs to be modified in order to perform optimally

OptimalSockets

- The number of sockets you should present to the VM to optimize it
- This number will match "VMSockets" if the VM is already configured optimally

OptimalCoresPerSocket

- The number of cores per socket you should present to the VM to optimize it

Priority

- The priority of the highest finding highlighted in details. The higher the priority, the bigger the impact the optimal configuration will have on the VM*

  *Note that performance improvements will vary depending on a number of factors specific to your environment.

Details

- Based on each finding, gives details on what should be done to optimize the VM performance. Note that there are some findings that need to be changed on the cluster or host level.

Priority	Detail	Explanation
Low	`VM does not span pNUMA nodes, but consider configuring it to match pNUMA architecture`	vCPU and memory fit into 1 pNUMA node, but best practice would be to match the host architecture.
Medium	`Host hardware in the cluster is inconsistent. Consider sizing VMs based on the minimums for the cluster`	Cluster has host(s) that do match others. VM findings are still based on current host, but if

		VMs can vMotion across hosts, the smallest host values should be used.
Medium	VM vCPUs exceed the host's physical cores. Consider reducing the number of vCPUs	Although hyperthreading enabled on a host can be beneficial, a VM should not be configured with vCPUs in excess of the number of physical cores on the host.
Medium	Consider changing the host Power Policy to "High Performance" for clusters with VMs larger than 8 vCPUs	Hosts with large VMs or IO latency sensitive workloads can benefit from the High Performance Power Policy setting. Performance Best Practices for VMware vSphere 6.7
High	VM <CPU/Memory> spans pNUMA nodes and should be distributed evenly across as few as possible	Virtual Machine vCPU and vNUMA Rightsizing – Rules of Thumb
High	VM has an odd number of vCPUs and spans pNUMA nodes	Virtual Machine vCPU and vNUMA Rightsizing – Rules of Thumb
High	VM spans pNUMA nodes, but pNUMA is not exposed to the guest OS: < • (vHW < 8) • (CpuHotAddEnabled = TRUE) • (vCPUs < 9). Consider modifying advanced setting "Numa.Vcpu.Min" to <#vCPUs> or lower • (Advanced setting ""Numa.Vcpu.Min"" is > VM vCPUs). The setting has been modified, but is still higher than VM vCPUs. Change the value to <#vCPUs> or lower to expose pNUMA to the guest OS" >	The host physical NUMA will not be exposed to the guest's vNUMA in the following cases: • vHW version is < 8 • CpuHotAdd is enabled on the VM • # VM vCPUs < 9, and is < Host cores/socket • Numa.Vcpu.Min setting is > # vCPUs (default is 9)