

HyperV_Mon 3.0

A Free tool from TMurgent Technologies

Version 3.0

Introduction

HyperV_Mon is a GUI tool for viewing CPU performance of a system running Hyper-V from Microsoft. Virtualization adds a layer of complexity to understanding true CPU usage of a virtual machine. This tool provides a simple to view set of graphs to depict actual processor utilization, including “overhead” processing due to the use of the Hypervisor.

HyperV_Mon recognizes and supports Hyper-V R1 (Windows Server 2008 R2), R2 (Windows Server 2008 R2 with SP1), and R3 (currently Windows Server 8 CP, but presumably also future versions).

HyperV_Mon is **not** intended to be a regular monitoring tool that you use every day. This tool is designed to help you understand what is going on when you need to know it. The tool requires that you have .NET 4.0 Framework installed on the machine that you wish to run the GUI on.

Typically you want to place this tool on a desktop that is not on the Hyper-V server to minimize impact on the server itself. For Server 2008 and 2008 R2, if you didn’t install Server Core for the host partition, you could place it on the root partition machine; On Server 8 you may place it on the Server Core host partition if you also install the GUI feature.

The tool leverages the Root Partition WMI provider – the same interface as is used by SCVMM and third party tools. So as long as the remote GUI user has WMI access to the Root Partition Server you are good to go.

Some of the new features in HyperV_Mon also need WMI access to the VMs themselves, but will reasonably operate if access to the VMs are not available.

Hyper-V Overhead

The term “**Overhead**” is used in this document and in the HyperV_Mon product as a shorthand term to capture the concept of processing that is being performed by the Hypervisor itself, rather than the workloads of the virtual machine partitions.

Our use of this term is not technically “overhead” in the sense of processing that would not be occurring if you didn’t virtualize.

While some of what we are calling overhead here is represents overhead, some of it represents processing that would have occurred in drivers of the VM in a non-virtualized system.

A more technically accurate description would be processing that occurred in the hypervisor on behalf of the root partition or a VM. One way to think of it is having the hypervisor be a privilege mode like kernel mode is to user mode from within the Windows OS.

The tool has no installer. Just an exe with a configure button. The tool does require (domain) administrator privileges for some of the WMI access. All of the access by this tool is read-only. The tool is fully resizable.

Configuring the Tool

The tool has three configuration items, which are stored in the HKEY_CURRENT_USER hive of the Windows Registry.

After copying the exe onto a system, start it and mouse click the “Configure” button. The Configure dialog is shown in Figure 1.

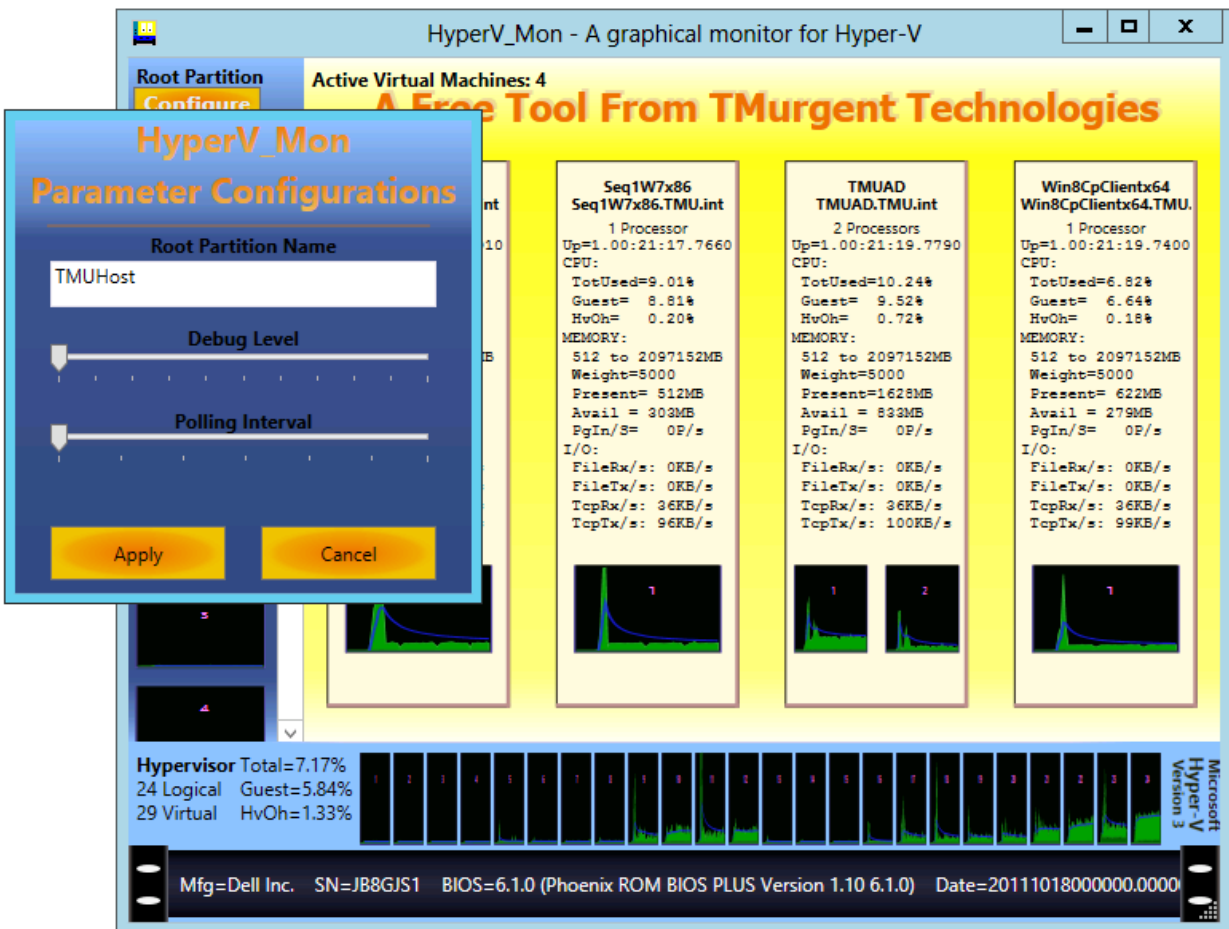


Figure 1 - The Configure Dialog

You must enter a qualified name for the host in the root partition of the server to be monitored. Any resolvable name may be used.

The Debug level defaults to 0 – meaning no debugging. You can move the slider to the right to enable debugging and configure how much information to log. Level 1 provides for errors only. The log file will be created or appended to in the current working folder.

The polling interval defaults to a one second interval. Move the slider to the right to slow down poling.

HyperV_Mon requires access to the WMI providers of the root partition of the target machine. This can prove to be a problem unless the machine (and user account you are running under) that the GUI is on is in the same domain as the Root Partition. See the blog of Microsoft’s John Howard <http://blogs.technet.com/jhoward/default.aspx> (especially Hyper-V Remote Management Parts 1-5) if you are having difficulty with this. Basically, if you can run C:\Windows\System32\Wbem\wbemtest.exe on your client machine and connect to the root partition machine you will be good to go.

Using the Tool

Figure 2 shows an example of the GUI in action.

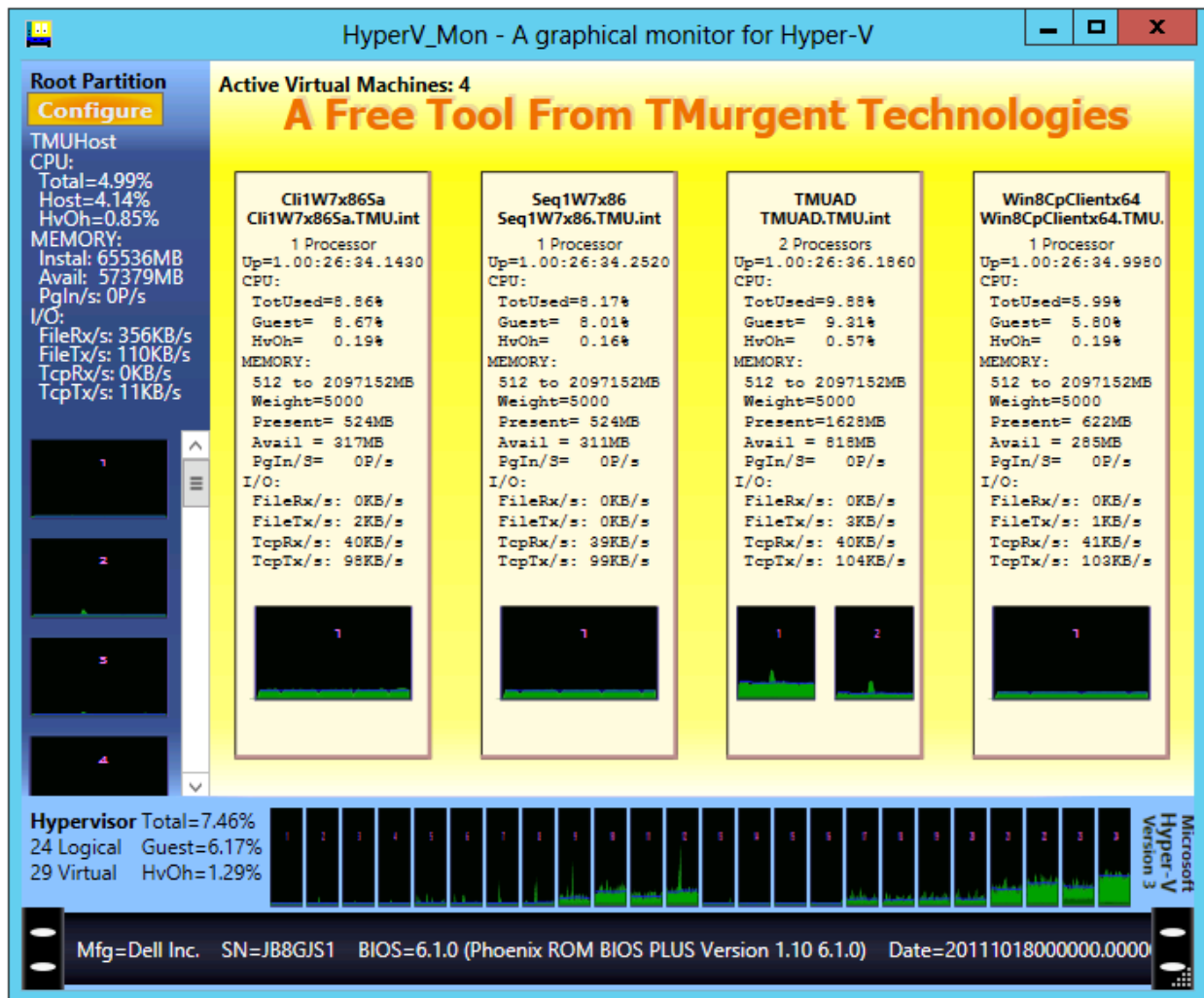


Figure 2 - HyperV_Mon in Action!

The tool consists of four basic areas; hardware, hypervisor, root partition, and guest virtual machines. Each are described in the following sub-sections.

Hardware

The bottom layer of the display represents the hardware. Not much to talk about there. It doesn't matter what your hardware looks like, the graphic will look like a 1U rack server from my lab. Some information is displayed identifying the hardware.

Hypervisor

The next layer up is the Hypervisor. Only Microsoft Hyper-V, Hyper-V R2 and Hyper-V R3 are supported (i.e. "not the other guys"). The hypervisor release version is now shown on the right of this layer.

The statistics for the latest measurement interval are shown on the left side of this hypervisor layer display. Total CPU, Guest, and Hypervisor CPU percentages are shown.

In the middle of the display are a number of graphs for the logical CPUs. The number of "Logical CPU" physically installed is detected and depicted, one per graph. Actual CPU usage at the physical layer is reported inside each graph. In our case, this was a system with two six-core processors with hyper-threading added to make 24 logical processors. In the figure, almost all of the CPU is related to Guest usage and is shown in a lighter shade of green. The last two logical CPUs are showing some CPU usage tied to Hypervisor calls, indicated in the darker green color at the bottom of the graph. It is easiest to think of the Hypervisor calls as "hypervisor overhead", although technically some of that "overhead" is processing that would have been performed the guest was on raw metal hardware. Each graph shows the last 60 interval measurements are displayed (1 second each, by default) with the most recent interval displayed on the right of the graph. The blue spline (curvy line) represents a running longer term average for trending (each dot of the blue line depicts the most recent 60-interval average total CPU at the time of the interval).

Root Partition

Above the Hypervisor layer the display has two major parts, situated left and right. The left holds the root partition OS (plus the configuration button described earlier).

The root partition OS is typically where you have Server Core is running. The name of the machine is shown, along with performance statistics from the latest measurement interval. CPU, Memory, and IO (file and network) are provided. The CPU statistics are divided between host OS (what you would see in the Windows Task Manager of the OS) and hypervisor overhead.

The Root Partition has "Virtual Processors", and Hyper-V will supply one for one for each "Logical Processor" in the Hypervisor. Each of these virtual processors has a graph depicting CPU measurements. When there are too many logical processors for this display, a scrollbar appears for them. The Hypervisor will schedule the Root Partition virtual processors (along with guest virtual processors) into logical processors as it sees fit. You can think of this as round-robin scheduling (with a tendency for a

thread to stay in a processor once scheduled) , although on larger systems like the one shown there may be a tendency to schedule to a physical processor to conserve power usage when loads are log (in the example shown, the second physical processor is represented by logical processors 7-12 and 19-24) . Note that if you ran a single threaded app on the root partition and used processor affinity within the Root partition OS, you would see all of the CPU for that app running in one of the root partition virtual processors on the left hand side of the display – just as you would see from the task manager inside the root partition. But looking at the Hypervisor layer you will see that the load was actually spread out among the logical processors!

Guest Virtual Machines

In the example this was a full blown 2008 with AD, DNS, IIS, Terminal Services, and a host of other things to be interesting

On the top right, in the yellow area, are the active Guest Virtual Machines. Machines that are not running are not shown.

The VM name and fully qualified OS name is shown at the top of each VM. The number of virtual processors allocated to the machine are shown below these names. Below this the length of time this VM has been running is shown (this counter comes from the hypervisor and may be different than the time since booting provided by the OS, as the counter is not reset when the VM resets).

Below this are the measurements from the latest interval for CPU, Memory, and IO for the VM. If Dynamic Memory is in use, the minimum and maximum memory settings are also shown.

Below this are CPU graphs showing the last 60 intervals of CPU usage for each virtual processor. As in the other CPU graphs, the most recent interval is on the right; light green represents CPU usage by the OS, and dark green represents hypervisor “overhead” for the VM.

Holding the mouse over a VM will reveal additional information that we pull from the Hypervisor. This information is not normally needed, but can shed some light on causes when the hypervisor misbehaves on certain hardware.

Holding the mouse over any of the CPU graphs in any section also reveals additional statistics.

Additional Information – Clock Slew

You should note that **the task manager inside guest OS machines lies**. This isn't the task manager's fault. In the example shown in Figure 3, the Windows Task Manager within the guest OS would be showing 100% CPU utilization where our snapshot shows only 96% (including the “overhead”. Microsoft refers to this as clock slew, but basically the Hypervisor is pulling the rug out from under the OS's nose (de-scheduling the virtual processor) and the OS doesn't know it. This is why we must pull information out of the hypervisor to know what is really going on.

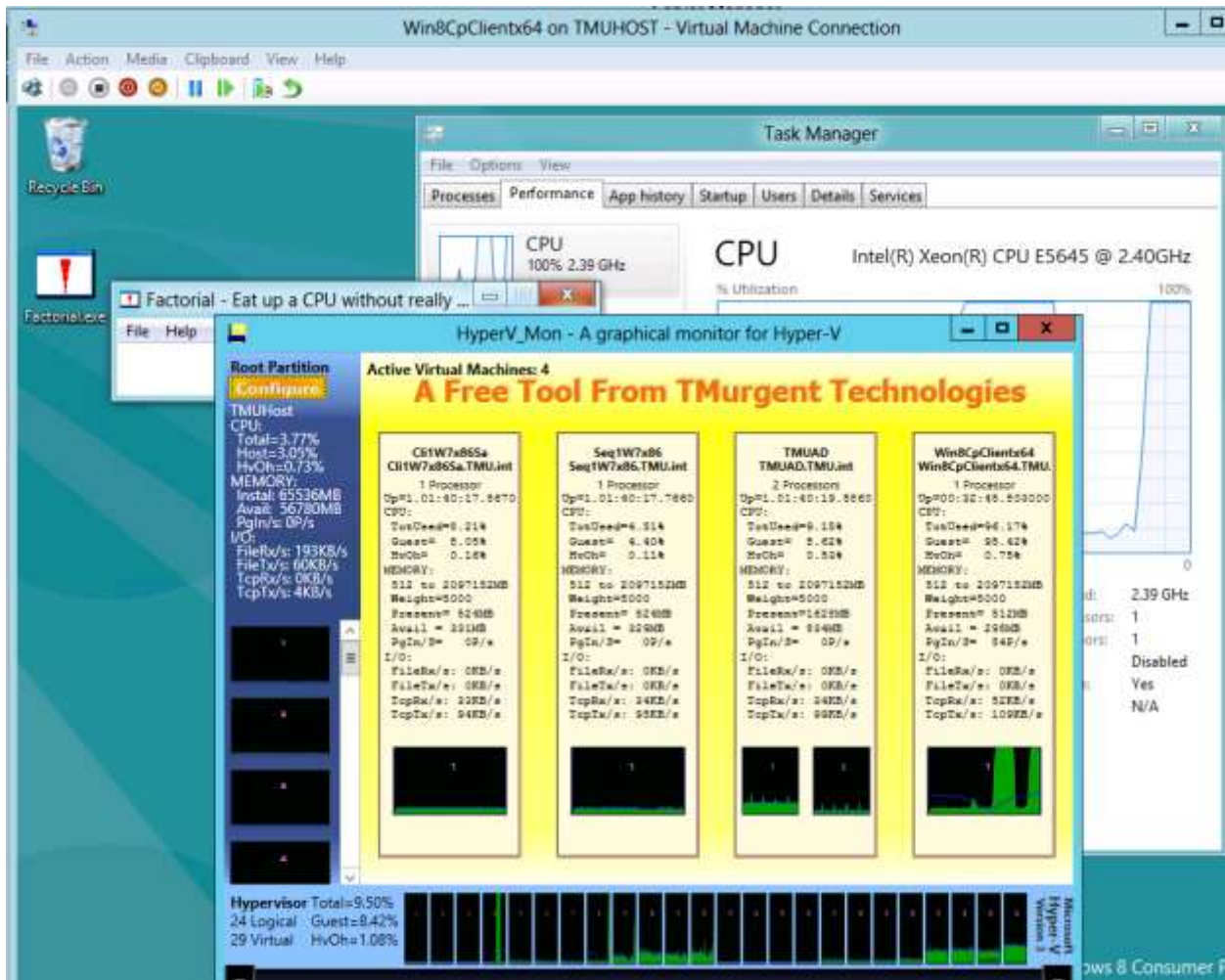


Figure 3 - Example of "Clock Slew"

Also note how the heavy processing inside of one VM is being spread between multiple logical processors on the Hypervisor layer.

Additional Information – Windows XP Performance

With this tool, it is easy to understand why Windows XP and earlier OSs significantly underperforms on Hyper-V in comparison to Windows Server 2003, or Vista, or above, VMs. You will see significantly more hypervisor "overhead" processing with the older operating systems because Microsoft made modifications to the newer operating systems to be hypervisor (and processor VT) aware. Those newer OSs do not display such behavior. It is all in the enlightenments! In a setup on some older hardware, it is common to see heavy hypercall CPU usage with heavy IO usage in the XP VMs as is seen in Figure 4 (using an older version of HyperV_Mon). The amount of overhead you see with Windows XP is not likely to be as high as is shown in this example. We never resolved the issue, but I believe that is an example of CPU/motherboard chip revision level that Hyper-V didn't handle well; this issue is why I made this tool in the first place!

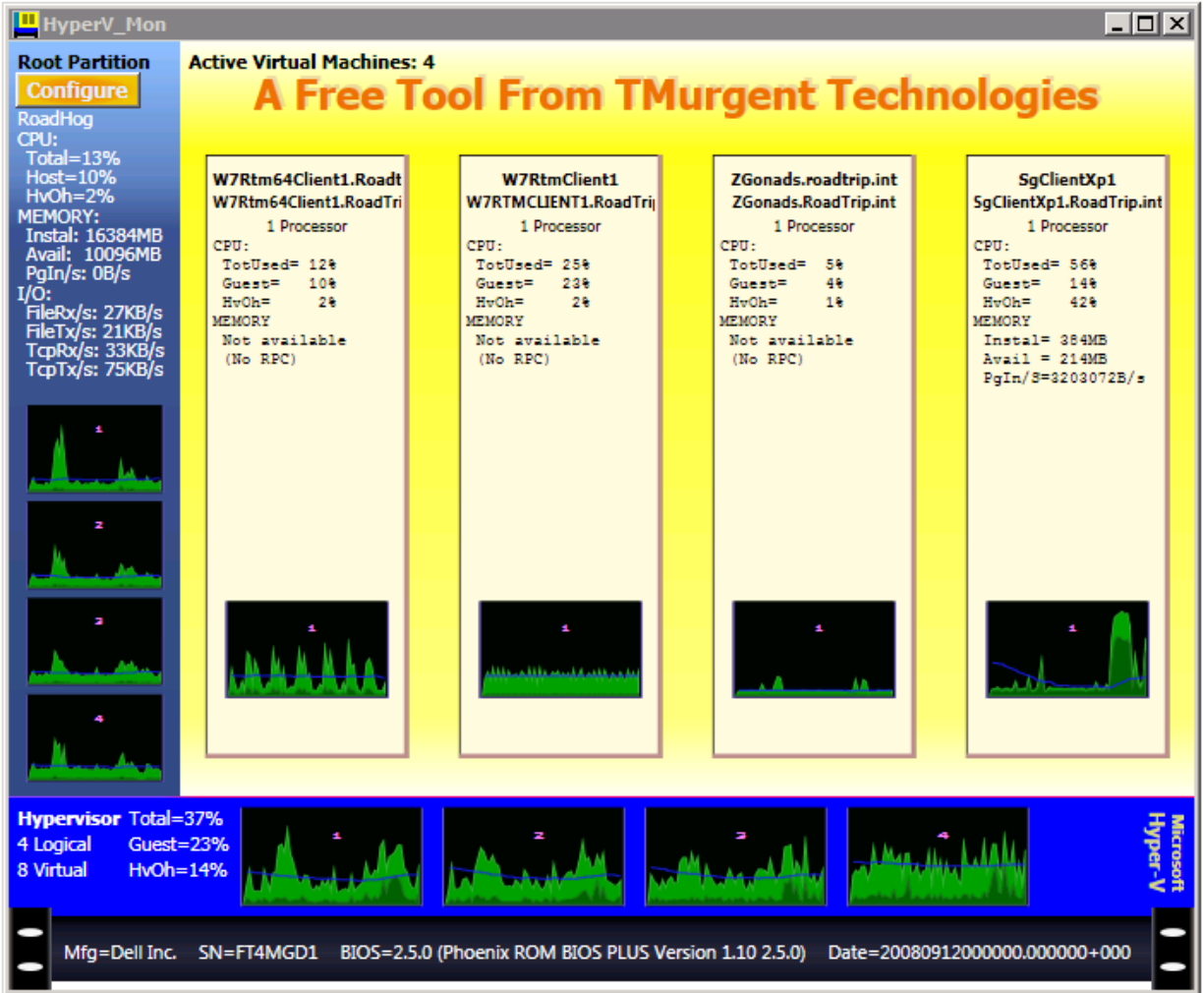


Figure 4 - Windows XP Performance

Legalese

The product is free to use for any purpose without license; has no support; is not warranted for any use. Feedback is encouraged. Gifts accepted gratefully.