**Maximizing VMware Performance and CPU Utilization**

*June 14, 2019* | [Susan Bilder](https://blog.heroix.com/blog/author/susan-bilder)

*In most environments ESXi allows significant levels of CPU overcommitment (that is, running more vCPUs on a host than the total number of physical processor cores in that host) without impacting virtual machine performance. (P. 20, ESXi CPU Considerations)*This post will discuss calculating CPU resources, considerations in assigning resources to virtual machines (VMs), and which metrics to monitor to ensure CPU overcommitment does not affect VM performance.

Our [**Overcommitting VMware Resources Whitepaper**](https://go.heroix.com/guidelines-for-overcommitting-vmware-resources) delivers the guidelines you need to ensure that you are properly allocating your host resources without sacrificing performance.

**Calculating available Host CPU Resources**

The number of physical cores (pCPU) available on a host is calculated as:

(# Processor Sockets) X (# Cores/Processor) = # Physical Processors (pCPU)

If the cores use hyperthreading, the number of logical cores is calculated as:

(# pCPU) X (2 threads/physical processor) = # Virtual Processors (vCPU)

For example, if you have 2 processors with 6 cores each:

(2 Processor Sockets) X (6 Cores/Processor) = 12 Physical Processors (pCPU)

(12 pCPU) X (2 threads/physical processor) = 24 Virtual Processors (vCPU)

Please note that hyperthreading does not actually double the available pCPU. Hyperthreading works by providing a second execution thread to a processor core. When one thread is idle or waiting, the other thread can execute instructions. This can increase efficiency if there is enough CPU Idle time to provide for scheduling two threads, but in practice [performance increases are up to a maximum of 30%](https://www.nersc.gov/assets/pubs_presos/CUG13HTpaper.pdf) and are strongly application dependent.

**Considerations in Allocating vCPUs to VMs**

**► Best Practices recommendations**

* Start with one vCPU per VM and increase as needed.
* Do not allocate more vCPUs than needed to a VM as this can unnecessarily limit resource availability for other VMs and increase CPU Ready wait time.
* The exact amount of CPU overcommitment a VMware host can accommodate will depend on the VMs and the applications they are running. A general guide for performance of {allocated vCPUs}:{total vCPU} from the Best Practices recommendations is:
  + 1:1 to 3:1 is no problem
  + 3:1 to 5:1 may begin to cause performance degradation
  + 6:1 or greater is often going to cause a problem

**► Non-Uniform Memory Architecture (NUMA)**

In a previous post on minimizing CPU latency with NUMA, we discussed performance degradation on multiprocessor VMs in which the number of vCPUs was greater than the number of vCPUs in a NUMA node.

**Generally, try to keep multiprocessor VMs sized to fit within a NUMA node.**

**► Co-stop**

The operating system on a VM requires synchronous progress over all of its vCPUs.  In order to keep track of the synchronicity of the vCPUs assigned to a VM, the hypervisor keeps track of the difference in progress for each vCPU assigned to the VM - this difference in progress is called the “skew”.  Skew values that are too high (typically over a few milliseconds) are an indication that the VM is unable to access all its processors synchronously.

Earlier versions of VMware (ESX2.x) used “[strict co-scheduling](https://www.vmware.com/content/dam/digitalmarketing/vmware/en/pdf/techpaper/vmware-vsphere-cpu-sched-performance-white-paper.pdf)”.  This kept track of a cumulative skew value over all the VM’s vCPUs and if the cumulative skew was over a threshold it would put the VM into a “co-stop” state which would stop the VM until there were enough physical CPUs to be able to schedule all the VM’s vCPUs simultaneously.  Since the co-stopped VM had to wait for enough physical processors to be available to accommodate all its virtual processors, strict co-scheduling could result in scheduling delays and idle physical CPUs.

Strict co-scheduling was replaced by “relaxed co-scheduling” in ESX 3.  Relaxed co-scheduling looks at per-vCPU skew values rather than looking at the cumulative skew values.  If a vCPU is making more process than its sibling vCPUs then its skew will increase and if the vCPU’s skew exceeds a threshold, that individual vCPU co-stops itself to allow its siblings to catch up.  Once the lagging vCPUs start making progress again, the co-stopped vCPU can co-start itself when a physical CPU is available.

Relaxed co-scheduling provided significant improvements in CPU utilization and made it much easier to scale VMs up to larger numbers of processors.  However, vCPUs can still end up in a co-stopped state and sizing a VM to use the minimal number of vCPUs that it needs will reduce the possibility of co-stopped vCPUs.

**Monitoring VMware CPU metrics**

[Monitor](https://go.heroix.com/virtualization-monitoring) the following metrics to fine tune the number of vCPUs allocated per VM and to ensure that CPU overcommitment does not degrade performance:

**► VM CPU Utilization**

Monitor CPU Utilization by the VM to determine if additional vCPUs are required or if too many have been allocated. CPU use can be monitored through VMware or through the VM’s operating system.

**Utilization should generally be <= 80% on average, and > 90% should trigger an alert, but this will vary depending on the applications running in the VM.**

**► VM CPU Ready**

VM CPU Ready is a measure of the time a VM has to wait for CPU resources from the host.

**VMware recommends CPU ready should be less than 5%.**

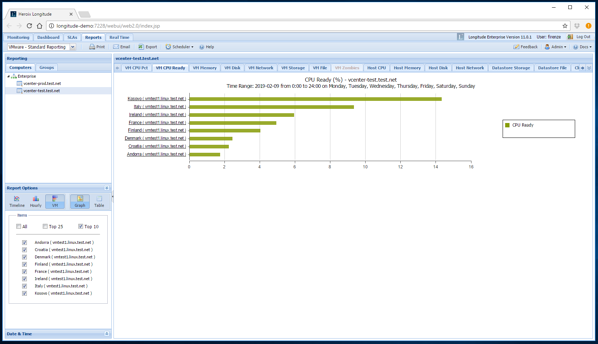
[](https://cdn2.hubspot.net/hubfs/2852520/Blog%20-%20Maximizing%20VMware%20Performance%20and%20CPU%20Utilization/Longitude%20CPU%20Ready.png)

Figure 1: Longitude Report of VMware showing high CPU Ready values.

**► Co-Stop**

Applicable to VMs with multiple vCPUs – is a measure of the amount time after the first vCPU is available until the remaining vCPUs are available for the VM to run.

**A co-stop percent that is persistently >3% is an indication that a right-sizing exercise may be in order.**

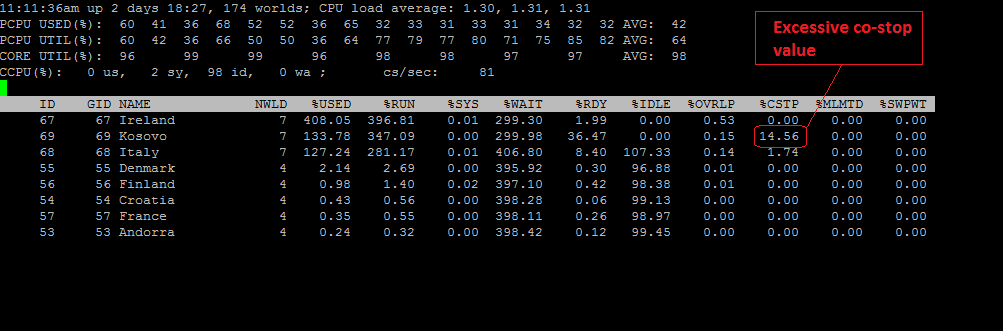
[](https://cdn2.hubspot.net/hubfs/2852520/Imported_Blog_Media/fig2.png)

Figure 2: esxtop showing a VM with a high co-stop value.

**► VMware host CPU Utilization**

Monitor CPU Utilization on the VM host to determine if CPU use by the VMs is approaching the maximum CPU capacity.

**As with CPU usage on VMs, CPU utilization at 80% - 90% should be considered a warning level, and >= 90% indicates that the CPUs are approaching an overloaded condition.**

[](https://cdn2.hubspot.net/hubfs/2852520/Blog%20-%20Maximizing%20VMware%20Performance%20and%20CPU%20Utilization/Longitude%20Capacity%20Planner.png)

Figure 3: Longitude Capacity Planner showing host CPU approaching capacity.

**Summary**

Overcommitting CPU allows you to maximize use of host CPU resources, but make sure to monitor overcommitted hosts for CPU use, and CPU Ready and Co-stop percentages.

Avoid oversizing VMs with more vCPU’s than needed. Consider NUMA architecture and the effect of co-stop waits when creating VMs with multiple vCPUs.