Correctly Sizing Your VMS: CPU, Memory and Storage Metrics to Consider

This is the first in a series of two whitepapers that show you how to correctly size your VMs to increase the number of VMs on existing hardware while assuring robust performance

Whitepaper by

Chris Chesley

Table of Contents

Introduction

This is part 1 of a two part paper designed to help you increase the numbers of VMs on your existing hardware. In this paper we discuss the CPU, memory and storage metrics that need to be considered in order to correctly size your VMs while assuring robust performance. In the second paper we present a process which assures your VMs are correctly sized.

Shared resources

It all starts with a clear understanding of your shared resources. There is a limited amount of memory, CPU and storage available at the host, cluster, resource pools or datacenter level. Any virtual machine using significantly more resources than planned has the possibility of impacting performance at any of these levels. If, for example, a host has significantly less memory than what is actually required by its guests you will have a performance bottleneck. Conversely, every VM using significantly less CPU, memory and disk than provisioned is wasting valuable resources.

When a VM is first deployed, we often over provision it to assure that it works properly. Once deployed, though, we rarely have the time or tools needed to go back and see what it is actually using. Before attempting to correctly size our VMs we need to appreciate a few subtleties related to our shared resources that can affect performance.

Memory

Applications typically release CPU and disk resources when they are not required. However, there are some applications and operating systems that improve performance by holding onto memory. This is one of the reasons that memory is usually the first bottlenecks that most administrators encounter first. The other issue that I have seen in many VMware environments is limits and reservations (these settings are not available in Hyper-V). Limits and or reservations can negatively impact the performance of virtual machines if not used with discretion and for a specific purpose. It is a best practice to avoid limits or reservations unless they are specifically required. If many people have the ability to deploy VMs, it is a good idea to regularly check that any reservations or limits set are actually required.

Ballooning measures how much memory the hypervisor is taking from a VM to keep it under a limit or to be reallocated. In a well run environment that has enough memory to go around, this value will be 0. A value over 100% indicates that a VM has a limit preventing it from getting all the memory it needs to run. Ballooning is a VMware only statistic and is not found in Hyper-V.

Swapping measures how much memory the host is forced to push to because of a memory shortage. This metric should also be close to or at 0 in an environment that is performing well and there is enough memory for all the virtual machines.

TPS (Transparent Page Sharing) measures how much read only memory is shared among virtual machines in a VMware environment and is not available in Hyper-V. These memory pages are typically operating system or application code and the more you can group similar operating systems or applications on the same host, the more TPS you will get which will reduce the overall memory usage. TPS will recognize any pages of memory in use that are used by multiple systems, loading only that page of memory once, and then pointing the guest machines to that page.

CPU

CPU and memory utilization are the most common constraints that will prevent deployment of higher numbers of VMs. While utilization numbers may be low, the key metric in VMware here is CPU Ready. This is a measurement of how much time a virtual CPU is waiting for space to run on a physical CPU. If there are too many virtual CPUs waiting for the same core, the virtual machine will be slow, inconveniencing end users.

There are two straight forward solutions to CPU contention issues: either move virtual machines around so that all the guests on a host or in a resource pool are sharing fairly, or set limits so that less important systems have a lower portion of the CPU time available. Setting affinity and anti-affinity rules for important virtual machines allows administrators to lock them down to a specific core, and then keep other guests from using that processor. Once affinity is set, VMotion and DRS (Distributed Resource Scheduling) is no longer possible in a VMware environment.

Disk I/O

Disk I/O is increasingly becoming the principal constraint as larger hosts (quad socket with 4-6 cores per socket and 256 GB of RAM) become more common**. IOPS (Input/output Operations per Second) are not available as counters or metrics from VMware or Hyper-V**. Therefore, disk command latency, disk queue latency and throughput (a total of disk reads and disk writes) are important to watch.

Disk command latency is the average amount of time it takes to complete a SCSI command. Disk queue latency measures how much time a command spends in the VMkernel queue. A good value for queue latency is zero is a well performing infrastructure. There is no set value for disk command latency since there are so many different types and speeds of storage.

The best way to detect under or over utilization is to continuously monitor these statistics looking for significant change. It is also vital to view disk I/O from a datastore perspective and see how much traffic each host and virtual machine is using to determine if a particular guest or host is causing a bottleneck so it can be moved to another LUN, resolving the bottleneck.

Measuring Your Key Metrics

The first step in increasing the VM density is to actively monitor the above metrics in your infrastructure. You will need to monitor each VM individually to determine how much of the CPU, memory and disk originally provisioned are actually being used. You will be looking for under-provisioning that causes performance bottlenecks as well as over-provisioning which wastes resources.

For example, assume there is a virtual machine that has 2 CPU's assigned for it to use. It is checked after a week of utilization and found to be using only 7.1% of those CPUs at the peak value, and 2.3% average utilization. This means that the CPU on that system has been over allocated. That virtual machine should be reconfigured to use just 1 CPU, leaving a core open that can be used by other virtual machines. The performance implications of this are that assigning more CPUs than are necessary can slow down the execution of the VM since the host has to find that amount of cores open at the same time. The more cores it needs to find available at the same time, the longer it can take to get the required number of CPUs.

This same type of analysis should be done for memory and storage as well.

Automating the Process

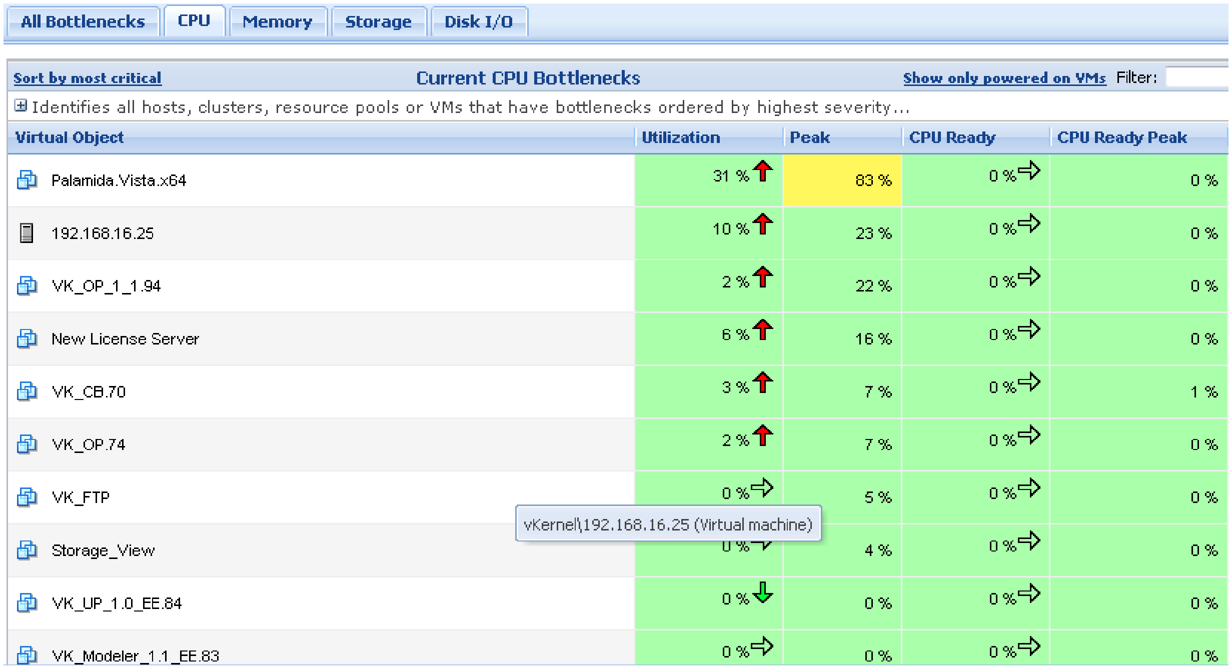
If you have a small, relatively static environment, this process will be manageable and may not require automation. If, however, the task of collecting these statistics regularly appears laborious, you should consider an automated solution such as VKernel's Optimization Pack and Capacity Analyzer products.

Sort by most critical — **Current Memory Bottlenecks** — Show only powered on V

⊞ Identifies all hosts, clusters, resource pools or VMs that have bottlenecks ordered by highest severity...

Virtual Object	Utilization	Peak	Swapped	Swapped Peak
OracleXP	77 % ⇩	79 %	25 % ⇨	25 %
T_UP1.2(RT)	78 % ⇧	78 %	1 % ⇨	5 %
SQL2008	93 % ⇧	93 %	1 % ⇨	1 %
T_CA4.2SE_fromTemplate	90 % ⇧	90 %	0 % ⇨	0 %
VK_UP_1.1_GA_115.114_(5915)	89 % ⇧	89 %	0 % ⇨	0 %
aa_opensuse_archivarepo	88 % ⇧	88 %	1 % ⇨	1 %

Capacity Analyzer automatically collects CPU, memory and disk utilization statistics, computes peak, average and significant changes and compares the data to provisioned levels. It automatically identifies current and future performance bottlenecks as well as the ability to add additional VMs.



All Bottlenecks | CPU | Memory | Storage | Disk I/O

Sort by most critical — Current CPU Bottlenecks — Show only powered on VMs Filter:

Identifies all hosts, clusters, resource pools or VMs that have bottlenecks ordered by highest severity...

Virtual Object	Utilization	Peak	CPU Ready	CPU Ready Peak
Palamida.Vista.x64	31 %	83 %	0 %	0 %
192.168.16.25	10 %	23 %	0 %	0 %
VK_OP_1_1.94	2 %	22 %	0 %	0 %
New License Server	6 %	16 %	0 %	0 %
VK_CB.70	3 %	7 %	0 %	1 %
VK_OP.74	2 %	7 %	0 %	0 %
VK_FTP	0 %	5 %	0 %	0 %
Storage_View	0 %	4 %	0 %	0 %
VK_UP_1.0_EE.84	0 %	0 %	0 %	0 %
VK_Modeler_1.1_EE.83	0 %	0 %	0 %	0 %

vKernel\192.168.16.25 (Virtual machine)

Conclusion

While this process can be done without an automated solution, it can be very time consuming and tedious. VKernel products automate capacity management processes. The value of eliminating waste and "correct sizing" virtual machines has allowed our customers to reclaim 20% to 35% of existing resources for use in deploying more VMs. This has allowed administrators and managers to make better use of their existing systems and as a result delay purchasing more servers and storage.