Software

vCPU Sizing Considerations

Accurately Determine how many vCPUs to Allocate to Virtual Machines

Written by David Davis, Virtualization Evangelist & Alex Rosemblat, Product Marketing Manager

Abstract

The number of vCPUs is one of the most important considerations when sizing virtual machines. But getting the right balance—neither over-allocating nor under-allocating—is a challenge. This white paper offers the guidance you need to get vCPU sizing right.

Introduction

vCPU sizing is one of the most important considerations when sizing virtual machines

When sizing virtual machines, a virtualization administrator must select the number of vCPUs, the size of the virtual disk, number of vNICs and the amount of memory. Out of all those potential variables, the two most difficult to determine are always the number of vCPUs and the amount of memory to allocate. This is because CPU and memory are the most finite resources that a server has, and these resources are also the most dynamically demanded resources by the guest OS in each VM. For a discussion of virtual machine memory sizing, see "VM Memory (vRAM) Sizing Considerations"; here, we'll cover proper vCPU sizing in your virtual machines.

The challenge: maximizing server hardware ROI while ensuring high application performance

Virtualization administrators should avoid over-allocating vCPUs because doing so wastes expensive server resources and will minimize ROI on that infrastructure. In fact, over-allocation of vCPUs in some VMs will actually cause performance problems for that VM and other VMs. On the other hand, the business critical applications running in virtual machines also need to maintain high performance and they need processing power to do so. The last thing a virtualization administrator wants is to have performance complaints from end users. VM administrators face a difficultly in balancing the need to maximize ROI of the server hardware and the requirement for applications to perform optimally.

Fortunately, with the right tools in place and the guidance from this whitepaper, virtualization administrators will be able to easily make the right decisions about vCPU sizing.

Only multi-threaded applications will benefit from having more than one CPU.

Understanding CPU usage in virtual servers

Physical and virtual servers are different when it comes to CPU usage. With the traditional physical server or desktop PC, you have an entire CPU or multiple CPUs (each with multiple cores) dedicated to the OS and applications running on it. The virtualization hypervisor adds an additional layer between OS and the physical CPU, allowing multiple virtual machines to share the hardware. Instead of the CPU requests from applications going to the OS and then the OS scheduling them on the physical CPU, the OS in the VMs talks to virtual CPUs (which it thinks are real physical CPUs). Requests from the multiple virtual CPUs are scheduled, by the hypervisor, across the multiple physical CPU cores. Just like with memory sharing in virtualization, with CPU sharing in virtualization, there is the traditional OS CPU scheduler and the hypervisor CPU scheduler.

All of this enables greater utilization and massive sharing of the server's physical CPUs.

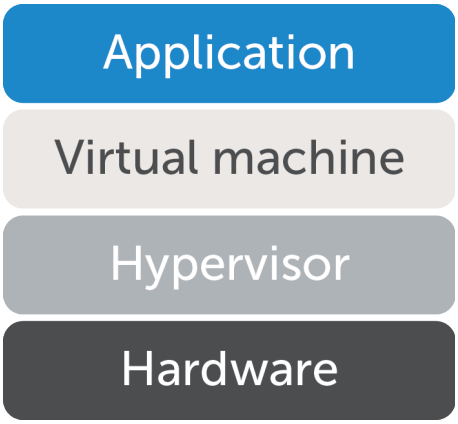

Figure 1. The virtualization stack

To summarize, the difference in CPU usage with physical and virtual servers is this:

- **In the physical world**, applications are scheduled by the OS onto the physical CPUs.
- **In the virtual world**, applications running on each OS, in each VM, make requests of virtual CPUs that are then scheduled by the hypervisor.

Adding CPUs benefits multi-threaded applications only.

Not only do today's servers have multiple CPU cores in each CPU, but they also use multi-threading (known as "hyper-threading" if you are using an Intel CPU). This means that each CPU core can execute "threads" in parallel. You can think of a thread as a process, so envision a CPU core being able to run multiple processes all at the same time (in parallel). However, having more than one CPU will benefit only applications that are multi-threaded. In other words, dedicated task-based servers (such as database servers and web servers) that run a single-threaded application will see no performance benefit by adding more CPUs.

Determining whether an application is single-threaded or multi-threaded
To know if an application is single-threaded or multi-threaded, you have two options:

- Ask the software manufacturer if the application is multi-threaded and supports SMP (symmetric multi-processing).
- When you have more than one processor or core on a server, see whether the program's multiple processes are using

more than one processor or core at a time. For example, if you have a quad-core CPU, watch the application as the CPU demands increase. If the application can use only 25 percent of total CPU capacity (one of the four cores), then it is a single-threaded application that can't use more than one core. On the other hand, if the application is using 50, 75 or 100 percent of the total CPU capacity (four of the four cores), then it is multi-threaded.

Whether you are running an application on a physical server or a virtual machine, you want to have multiple CPUs available only if the application running on that host can take advantage of those CPUs with multiple threads.

Having multiple vCPUs when they are not needed will slow down VMs.
On a physical server, if you have multiple CPUs available for the server's primary application but that app isn't multi-threaded, you are wasting the cost of those CPUs. However, in a virtual infrastructure, if you over-allocate multiple vCPUs to a VM and the primary app on that VM isn't multi-threaded, you could actually cause performance issues for that VM and others. This is because, with multiple vCPUs, the hypervisor's CPU scheduler must wait for multiple physical CPU time slots to become available before it can process requests from the multi-vCPU VM.

In other words:

- **With one vCPU**, CPU requests are quickly processed (or they are waiting on pCPU if no pCPU is available), as illustrated in Figure 2
- **With multiple vCPUs**, the hypervisor CPU scheduler must wait for multiple pCPUs to be available, as shown in Figure 3. Therefore, having multiple vCPUs when they are not needed will slow down VMs.

If your virtual hosts don't have the CPU capacity that today's applications demand, you should solve that problem before analyzing vCPU allocations.

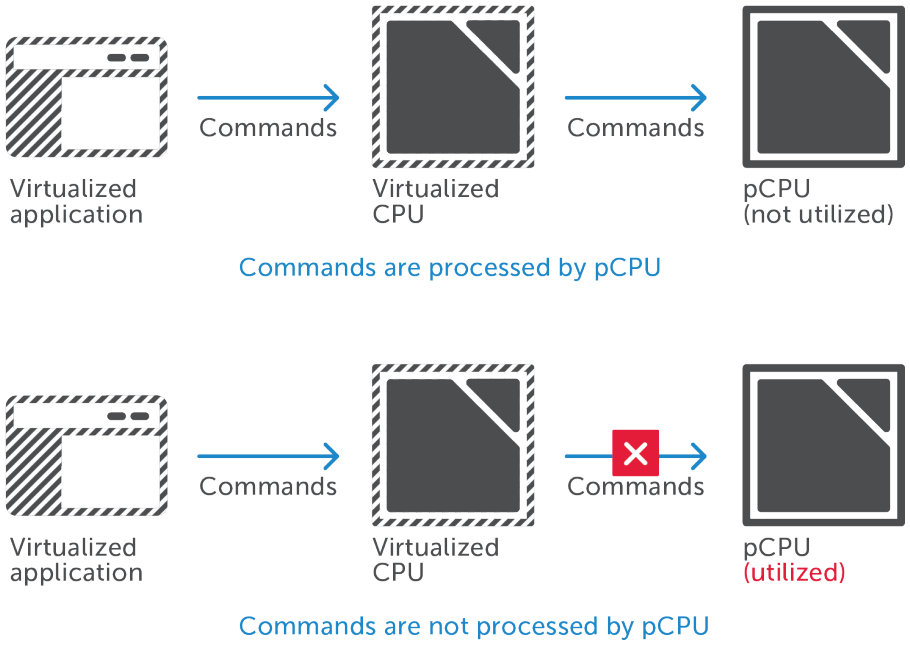

Figure 2. Application command processing with one vCPU

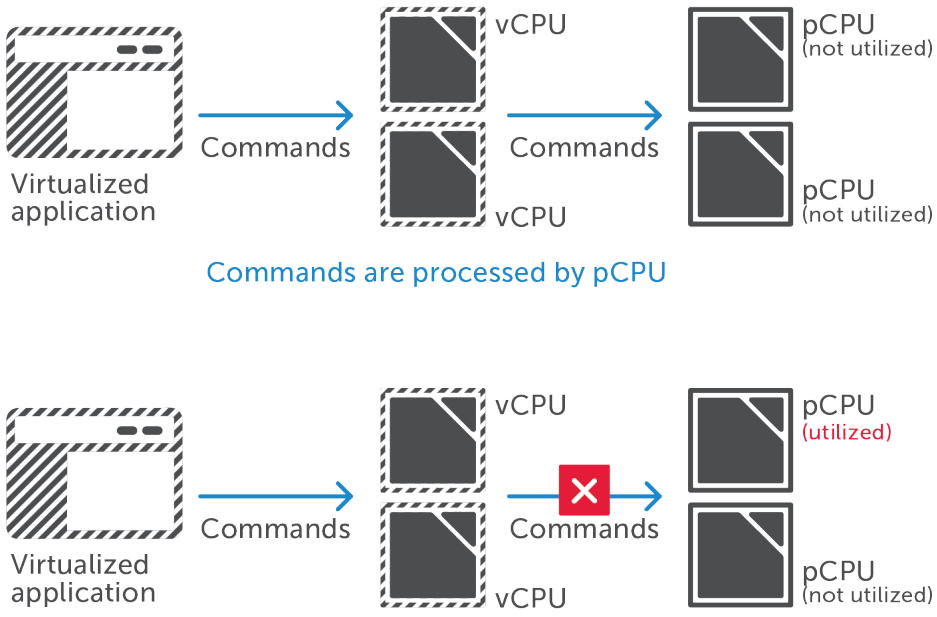

Figure 3. Application command processing with multiple vCPUs

100 percent CPU usage for short periods isn't cause for changing CPU configuration. Instead, look at CPU statistics over a slightly longer sample time to ensure that you are seeing trends instead of instantaneous usage.

Things to know before you begin analyzing vCPU allocations

Before you begin your analysis, resolve any virtual host and VM performance problems.

Deciding the proper number of vCPUs for a VM should be a long-term performance planning exercise that you perform periodically. Before starting that vCPU planning exercise, you should first determine if there is already CPU over-utilization in your virtual machines or on your virtual host. Typically, this is done by checking the following:

- **CPU ready (cpu.ready.summation) in milliseconds (ms)**—This statistic is available per virtual machine or per vCPU. It shows the number of milliseconds that the VM was ready to execute requests on the virtual host's CPU but there was no pCPU available to do so. An increasing CPU Ready value for a VM indicates that there is an ongoing lack of CPU cores on the virtual host.
- **CPU usage (cpu.usage.average) in percent**—This CPU usage statistic is available per host, per VM or per resource pool. It shows the percentage of time that the CPU was utilized, over the time range selected. Note that the total CPU includes all vCPUs on a VM or all pCPUs on a host. If you have greater than 80 percent CPU utilization on a VM or host over a one-hour time period, you should find ways to solve that problem. Those solutions could include reducing the number of vCPUs on your VMs, migrating active VMs to another host, adding more pCPU to a host, or dealing with misbehaving applications.

If your virtual hosts don't have the CPU capacity that today's applications demand, you should solve that problem first. It should be noted that one of the potential solutions for solving high virtual host CPU utilization is to reduce the number of vCPUs allocated to VMs if they were dramatically over-allocated.

VMware performance metrics to use for virtual host pCPU and VM vCPU usage analysis

Once you've ensured that your physical server isn't over-utilized, it's time to move on to a structured analysis of vCPU allocation on one or across

all virtual machines. We'll provide a detailed process below but first, let's cover two statistics you need to know before you start that process:

- **CPU usage maximum (cpu.usage.maximum) in percent**—This shows the maximum amount of CPU that was used at any one time, as a percentage. Keep in mind that if you have two vCPUs, a 50 percent utilization value is 100 percent of one vCPU and 0 percent of another, or 25 percent of one and 25 percent of another. This value will be used to determine if we hit the maximum amount of CPU already allocated to that VM during the time range.
- **CPU usage average (cpu.usage.average) in percent**—This value shows the average CPU utilization over the time range. Note that this is based on the total number of vCPUs; for example, if you have two vCPUs, a 100 percent utilization over the time range is 100 percent of both vCPUs.

Understanding time ranges and sample periods

To be able to accurately understand statistics that you are viewing, you must fully understand that these stats are measured over the default sample periods or whatever sample period you specify. For example, if you measure real-time or one-minute CPU usage, it is very likely that you will see periodic CPU utilization peaks of 100 percent. Short periodic CPU usage at 100 percent isn't cause for alarm or changes to CPU configuration. Instead, I recommend that you look at CPU statistics over a slightly longer sample time to ensure that you are seeing trends instead of instantaneous usage.

Some number of the statistical samples are pulled out and stored in a historical database for use when you run a performance report over a specific time period. Understanding both the sample time and the time range are important when it comes to interpreting performance graphs.

How to view and modify vCPUs for VMs

Viewing and modifying vCPU configurations on your VMs isn't something VMware admins need to do every day; however, you do need to know where to do it and when you can make changes:

- **To view VM vCPU configuration**—Click on a VM and looking at the Summary tab.
- **To edit VM vCPUs**—Go to a VM's properties and then click on the CPU in the hardware tab.
- **To view vCPU configurations across all VMs**—Go to a higher level in the virtual infrastructure (from a VM, go up to the ESXi server or up to the resource pool or cluster level) and then click on the Virtual Machines tab.

Additional factors to keep in mind when allocating VM CPUs

Limits

By configuring a limit on a virtual machine, you are placing an artificial cap on the maximum amount of CPU that the VM can use. Without a limit, the maximum CPU that can be used is the Mhz of the pCPUs on the server, per vCPU allocated to a VM. Keep in mind that while a VM has full access to the total Mhz of a pCPU (per vCPU allocated), those pCPUs are still shared between that VM and all others on that virtual host. Note that limits configured in the hypervisor aren't visible to the guest OS, which can further cause unexpected application performance issues.

While functionality exists to configure a Mhz limit on a virtual machine's vCPU, it rarely makes sense. Instead, it makes more sense to set a CPU Mhz limit across all VMs in a particular resource pool.

Many times, VM CPU limits are put in place by an administrator who did not fully understand the poor performance and wasted resources that limits can create. When looking at CPU utilization, CPU limits skew performance metrics and can cause confusion while

You can configure a Mhz limit on a virtual machine's vCPU, but it rarely makes sense. Instead, set a CPU Mhz limit across all VMs in a particular resource pool.

Take the time to understand how and when applications are used. Those insights will help you properly size vCPU for the VM, configure resource controls (if needed), and understand the priority of each application.

troubleshooting. In other cases, VM CPU limits have also been put in place by the VMware admin to limit physical resource consumption by applications (often unbeknown to the application owner).

Reservations

CPU reservations artificially set the minimum amount of CPU that a virtual machine (or resource pool) has access to. Even though those CPU resources may not be needed by the VM now, a reservation pulls those CPU resources away from other virtual machines that may need it. Like limits, reservations are better set on resource pools instead of individual virtual machines since they can hurt the performance of other virtual machines and skew CPU metrics due to the artificial requirements being put in place.

The application's needs

Taking the time to look inside the virtual machine and analyze the CPU resources that an application uses can yield a great deal of information about that VM's CPU needs. When evaluating a Windows operating system, you can run Resource Monitor and Performance Monitor to expose which processes use the most CPU and how it varies.

Also, speaking to the business owner for an application helps determine when that application is used, who uses it, what would happen if it were unavailable, and how the use of the application is growing (or shrinking).

By taking time to understand these factors, VM administrators can draw further insights into properly sizing vCPU for the VM, configuring resource controls (if needed), and understanding the priority of the application as compared to other apps.

Special considerations when changing vCPU configurations

When you determine that the number of vCPUs that a VM has configured needs to be changed, there are a few considerations:

- When going from one vCPU to many, or from many to one, vCPU kernel changes are required in the VM guest operating system. For example, with Windows Server 2003, you need to make a change to the HAL (see VMware KB article 1003978 for more information). However, with Windows Server 2008, you can switch between single and multiple CPUs without making any HAL changes.
- Virtual machines need to be shut down to remove vCPU, and most VMs need to be shut down to change vCPU unless the vSphere "hot plug CPU" feature was enabled before boot AND the VM meets the operating system requirements to use that feature.

The VM CPU sizing process

Overview

Not only is a virtual environment dynamic, but the usage of the applications in the VMs will be in constant flux. As a VMware admin working on a critical production virtual infrastructure with hundreds or thousands of constantly changing virtual machines, you need a formal process for proper vCPU sizing, not just a "rule of thumb."

This process, ideally, involves the application admins and will have to be undertaken both when a new VM is being created for an application and on a periodic basis.

The workflow diagram on the next page introduces a best practice for how to execute this process. While this process may need slight modification for certain companies, it will work as is for most VMware administrators.

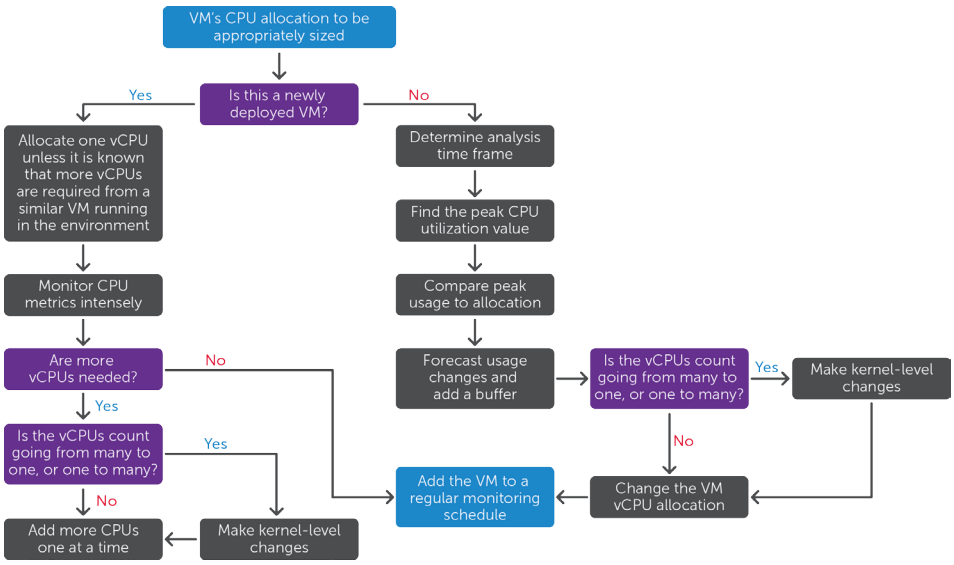

Figure 4. The VM CPU sizing workflow

The vCPU sizing process for an existing VM

Let's look at the steps in this vCPU sizing process if we are not dealing with a newly deployed VM (the branch to the right in the figure).

Step 1: Determine the time frame for analysis.
As I mentioned earlier in this guide, when performing your analysis, you need to take the perspective of the time frame into consideration. It's easy to interpret statistics incorrectly and make improper vCPU configurations by looking at a poorly selected time range.

While it is difficult to pick one time frame that is perfect for all situations, I generally recommend a time frame that is no less than one week. However, in some cases, that time frame may be as much as a year (such as a retail company that has high utilization around the holiday season or a university that has high utilization around registration). On the other hand, if you have a common application that you know is relatively predictable, you can safely look at time frames between one week and one month.

Step 2: Find the peak (maximum) value.
So what peak value do you choose? Peak values are interesting because they show whether the vCPUs were ever 100 percent utilized during your time period. However, you need to investigate exactly when the 100 utilization occurred (for example, was it during backups?) and weigh your findings with the average utilization. For example, an instantaneous value of 100 percent CPU utilization for a single vCPU VM over a week means nothing if it happened only once during the backup window and the average utilization is just 25 percent.

In other words, don't just look at the current value and use that. Use the tips above in determining a time frame to make sure that you identify the true peak.

Make sure you identify the true peak by investigating occurrences of 100 percent CPU to see if they are one-time anomalies.

Be sure to provision sufficient vCPU for both current and expected future usage; otherwise, the VM may face vCPU-related performance issues later on.

Step 3: Compare the peak usage value to allocated CPU.
Once you find the true average and peak (maximum) percent of vCPU utilized during your sample, compare that to the number of vCPUs configured for the virtual machine. Then use these guidelines:

- If the average is less than 38 percent and the peak is less than 45 percent, consider downsizing vCPUs.
- If the average is greater than 75 percent and the peak is above 90 percent, consider adding vCPUs.

For example, if the VM is configured with two vCPUs and the maximum utilization is 100 percent, you are maxing out both vCPUs at some point. If the average utilization is consistently greater than 80 percent, then consider adding vCPUs.

On the other hand, if you find that that vCPUs never hit their maximum, average utilization is low, and you have more than one vCPU, then consider migrating the VM down to one vCPU (or reducing the VM's vCPUs by one).

Step 4: Forecast the expected growth rate and factor in a buffer to determine the right allocation.
Before you decide to downsize a VM's vCPU configuration, be aware that the vCPU demands of the VM (and, more specifically, its applications) could increase by the time that you repeat the vCPU sizing process. Ask yourself the following questions: Has the application owner been contacted? Was an increasing trend in vCPU demand noted based on historical analysis? It is important to provision sufficient vCPU for both current and expected future usage, or a VM may face vCPU-related performance issues.

Once you establish the expected growth rate, add a buffer to ensure that the vCPU allocation is accurate. While historical peaks have likely been taken into account, it is still good to factor in a buffer to ensure that the virtual machine's vCPU has headroom to grow into and does not max out. For less critical applications, a buffer may not be necessary, and can be avoided to retain more CPU capacity for other VMs. However, for multi-threaded critical applications whose CPU utilization fluctuates, adding an additional vCPU is recommended. This way, in the event that an unexpected, business-critical demand on CPU happens, end-user application performance won't suffer.

Step 5: Make kernel-level changes if needed, change the vCPU allocation, and document all your changes.
Once you've completed the first four steps, it is time to change the virtual machine's number of vCPUs.

First, shut down the guest OS. Unless the correct guest OS is installed and CPU hot plug is enabled (assuming you need to add additional vCPU), the guest OS must be shut down to add or remove vCPU from a virtual machine.

Next, resize the virtual machine's vCPU. Right-click on the VM (in the hosts and clusters inventory) and then click Properties. At that point, you'll see a window (see Figure 5) where you can resize the VM's CPUs on the Hardware tab.

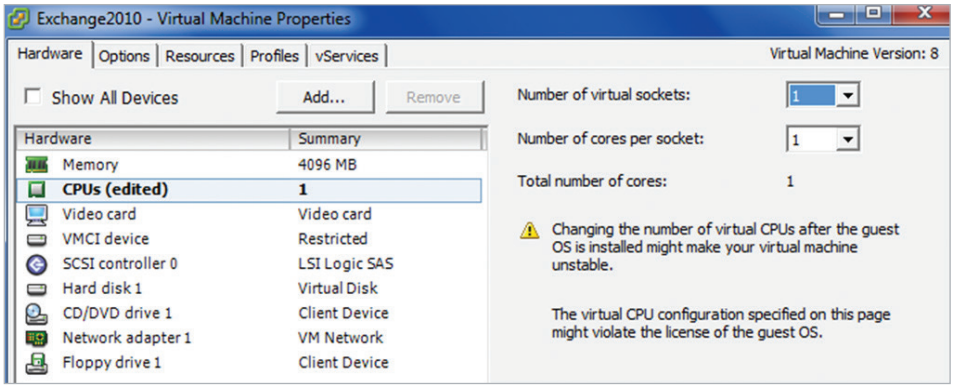

Figure 5. Configuring vCPU on a vSphere VM

Once the VM's vCPU has been resized, you need to take guest OS kernel and HAL changes into account. Typically, the servers that need HAL or kernel changes when going from one CPU to several, or from several to one, are the older Windows Server operating systems like Windows Server 2003. Without the proper changes, Windows 2003 Server systems will perform slowly or not at all. Windows Server 2008 and Linux-based systems don't require any HAL or kernel changes when their number of virtual CPUs is changed.

Finally, you can document the change to the VM's vCPU and power back on the VM.

The vCPU sizing process for a new VM

Now, let's look at the vCPU sizing workflow in Figure 4 again but this time, let's say that a new virtual machine is being provisioned, so we'll follow the path to the left. There won't be any historical performance data for this new virtual machine. Because of this, the vCPU configuration made for this VM could be much less accurate and the VM will need to be monitored more closely.

Step 1: Initially, allocate the minimum vCPUs possible.
In many cases, there will be some idea of how many vCPUs a new VM should have based on the same application (or a similar app) running on another VM. But even if this information is available, the process of monitoring and reviewing described below should still be followed.

On the other hand, if a new VM is being created for a new application or use case, and this is no indication how many vCPUs should be allocated for that app and its users, you could just go with the recommendations as if the app were being installed on a physical server. Or, you could start with just one vCPU and see how it goes from there.

It may be tempting to go with a gut feeling (such as the SharePoint admin's preference that his VM to have six vCPUs) or to use the same number of vCPUs that a physical server is configured with. But with those methods, it's likely that the resulting allocation will either grossly overshoot the number of vCPUs needed or, even worse, underestimate the number of vCPUs required by the application and its users.

Step 2: Monitor CPU metrics intensely.
Once a new VM has been deployed with a given number of vCPUs, CPU usage should be monitored intensely. Just how frequent and in-depth this monitoring needs to be depends on how critical the VM is. Administrators may also be able to rely on vSphere alarms to alert them if the new VM hits high CPU utilization.
When monitoring, look at the vSphere VM CPU usage level and also look inside the guest OS to see how much CPU is in use and by what application. Is the VM maxing out the base quantity of vCPUs you configured? If so, then more vCPUs are likely needed.

Once you have deployed a new VM, monitor CPU usage closely to see if more vCPUs are needed.

Because application usage changes, you should regularly review vCPU usage to ensure that performance remains high.

Step 3: Add more vCPUs one at a time. If the new VM is running low on CPU capacity, add more vCPUs—one at a time. Remember, you want to find the right number of vCPUs for the VM, not just throw a bunch of resources at it and potentially cause performance problems later. When you add an additional vCPU, be sure to monitor very carefully.

Set up a regular CPU sizing review based on an appropriate time frame.

Because application usage changes, vCPU must be continuously evaluated to ensure that performance remains high through any additional changes in the dynamic virtualized infrastructure.

Accordingly, you should set up a regular vCPU review process at appropriate intervals. My recommendation is to hold reviews once per month. However, some organizations that are growing quickly may want to perform reviews more frequently. Once a VM sizing process has been conducted a few times on the virtual machines in an environment, the steps will become more familiar and administrators will be able to build on their experience to streamline the process. Whenever you add a new VM, once the vCPU usage has been stabilized and the new VM has an appropriate number of vCPUs, be sure to include the new VM in the periodic vCPU review process.

Keep SLA requirements in mind.

As administrators, we strive to give applications what they need for peak performance. However, in the real world, due to service level agreements or cost models, this isn't always possible. In these cases, we have to intentionally give fewer vCPUs than would be ideal or use resource controls like limits in order to give a VM (and its underlying CPU-hungry application) only what should be allocated for business reasons.

Thus, keep service level agreements in mind when analyzing vCPU allocations and realize that there are cases where we have to allocate a vCPU value other than what is optimal.

Auto-size vCPUs in your environment

You can automatically calculate vCPU allocation using the techniques described in this white paper with a 30-day free trial of Dell vOPS Server Standard™. The free trial of vOPS Server Standard™ installs as a virtual appliance in 20 minutes.

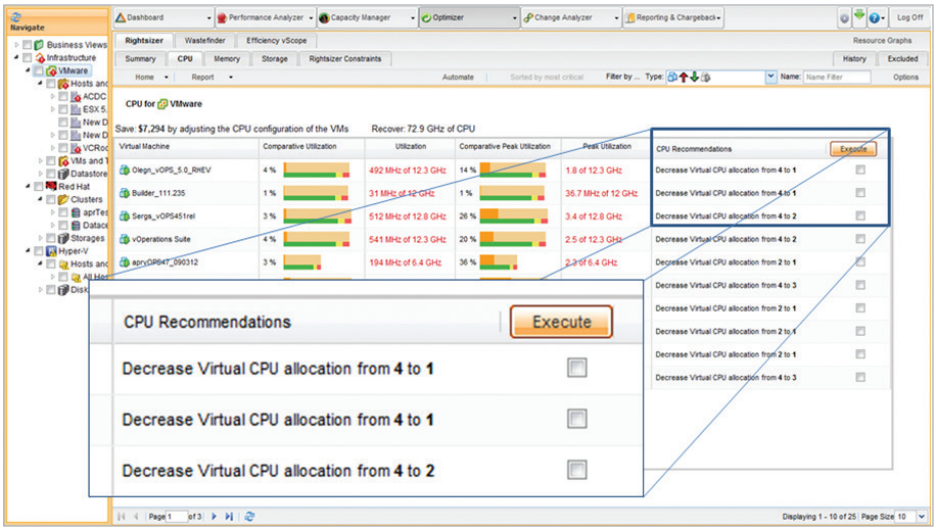

Figure 5. Automatically calculate vCPU with vOPS Server Standard™.

Conclusion

Besides memory, CPU is the most finite computing resource that a virtual infrastructure has. By implementing the workflow detailed in this paper, virtualization administrators can more efficiently use their computing resources. Data centers will be able to reclaim CPU resources for other VMs, enabling them to increase VM density and defer hardware purchases.

In additional, regular monitoring of CPU usage will help VM administrators proactively spot problem areas in VMs that are under-provisioned, so they can prevent performance issues. Ultimately, accurately sizing vCPU in virtual machines benefits everyone: the organization can realize a better return on investment for a virtualization initiative, and both application owners and end users see better application performance.

For More Information

About Dell

Dell Inc. (NASDAQ: DELL) listens to customers and delivers worldwide innovative technology, business solutions and services they trust and value. For more information, visit www.dell.com.

If you have any questions regarding your potential use of this material, contact:

Dell Software
5 Polaris Way
Aliso Viejo, CA 92656
www.dell.com
Refer to our Web site for regional and international office information.