VM Memory (vRAM) Sizing Considerations

Challenges and considerations in determining how much memory to allocate to virtual machines

WHITE PAPER BY DAVID DAVIS AND BRAD BONN

Table of Contents

Introduction 2

Memory Usage through Different Levels of the Virtualization Stack 3

Memory Ballooning 4

Transparent Page Sharing 5

Memory Compression 5

VMware Performance Metrics to use for VM Memory Usage Analysis 5

Determining if a VM has Memory-related Performance Problems 7

Additional Factors to Keep in Mind when Sizing VM Memory 8

"Memory Hog" Applications 8

Limits 8

Reservations 8

Knowing an Application's Needs 9

Instituting a VM Memory Sizing Process 9

Determining Time Frame for Analysis 10

Finding the Peak Value 11

Comparing the Peak Usage Value to Allocated Memory 11

Forecasting Usage Changes until Next Memory VM Allocation Review 12

Factor in a Buffer Value 12

Change the VM Sizing Allocation and Document the Change 12

Setting Up a Regular Memory Review Based on an Appropriate Timeframe 13

Provisioning a New VM for Appropriate Memory Allocations 13

Initially Allocate the Minimum Possible 14

Monitor Memory Metrics Intensely 14

Add More Memory in Small Increments 15

Add the New VM to the Regular Memory Sizing Review Process 15

Conclusion 15

About the Author 16

About the Sponsor 16

Introduction

When sizing virtual machines a virtualization administrator must select the number of vCPUs, the size of the virtual disk, the number of vNICs, and the amount of memory. Out of all those resources, the amount of memory allocated to each virtual machine (VM) is often the most difficult to determine. This is because memory is the most dynamic and least predictable of those resources.

Virtualization administrators should avoid over-allocating memory to their VMs because doing so can waste expensive server resources and will, therefore, minimize the return on the investment (ROI) in that infrastructure. On the other hand, the business-critical applications running on virtual machines also need to maintain high performance, and the last thing a virtualization administrator wants is to have performance complaints from end users.

VM administrators often face difficultly in balancing the need to maximize ROI of the server hardware and the requirement for applications to perform well. Fortunately, with VMware's memory over-commitment capabilities and the guidance from this whitepaper, virtualization administrators will gain a deeper understanding about how to make VM memory sizing more straightforward and less of a painful trial-and-error process. By reading this whitepaper, VM administrators will learn:

- How memory usage works at different levels of the virtualization “stack” (see Figure 1)
- The challenges in determining a VM’s memory requirements
- How to screen for memory-based VM performance issues before attempting to optimize an environment
- How to approach appropriate sizing of new VMs that need to be deployed
- How to institute a regular memory sizing process for the data center
- What VM performance metrics provide the best insights on memory usage and how to access these

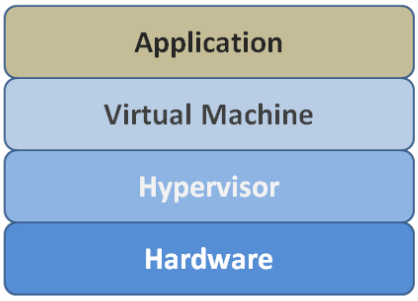

Figure 1 – The “Virtualization Stack”

Memory Usage through Different Levels of the Virtualization Stack

Physical memory on the host is managed by the hypervisor, and the hypervisor loads this memory just like any other operating system does. VMs loaded in the hypervisor will increase the memory utilization incrementally as users create virtual machines. Once the guest OS is running and the applications load, the guest OS believes that it owns all the virtual memory that has been configured (just as it does on a physical server) and will often "pin down" large blocks of memory to cache it for future use.

It is important to note that virtualization is not a panacea and there is memory overhead associated with using this technology. Guests and their applications will never have access to the same amount of physical memory installed on the host. There is an additional quantity of physical memory that is used up with each virtual machine powered-on in the hypervisor, and the amount of overhead is determined by the virtual resources that have been allocated to the VM (see figure 2). Larger amounts of virtual RAM and additional vCPUs increase the overhead memory amount on the host (see figure 3).

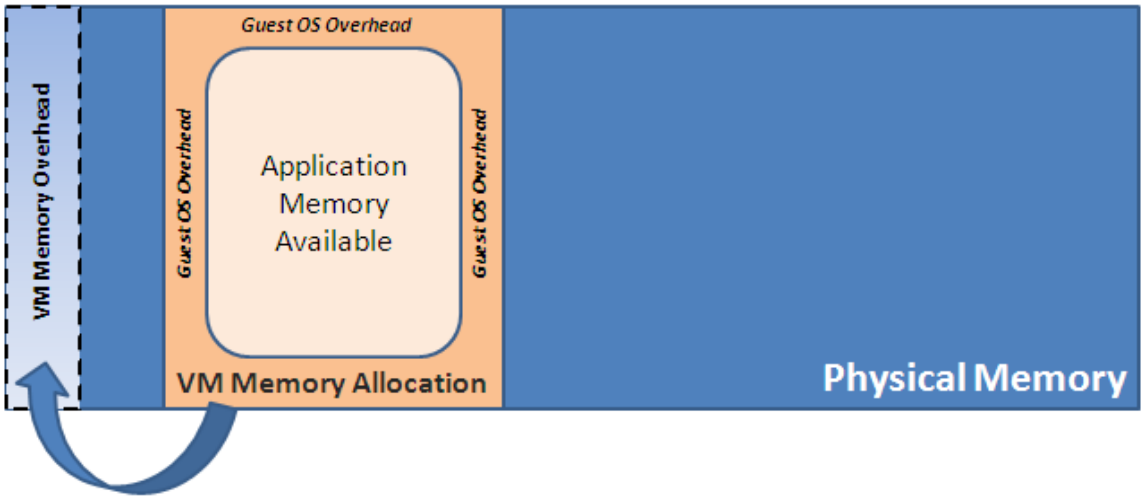

Figure 2 – The Impacts to Available Memory for an Application from VM and Guest Overhead

Table 3-2. Overhead Memory on Virtual Machines

Memory (MB)	1 VCPU	2 VCPUs	3 VCPUs	4 VCPUs	5 VCPUs	6 VCPUs	7 VCPUs	8 VCPUs
256	113.17	159.43	200.53	241.62	293.15	334.27	375.38	416.50
512	116.68	164.96	206.07	247.17	302.75	343.88	385.02	426.15
1024	123.73	176.05	217.18	258.30	322.00	363.17	404.34	445.52
2048	137.81	198.20	239.37	280.53	360.46	401.70	442.94	484.18
4096	165.98	242.51	283.75	324.99	437.37	478.75	520.14	561.52
8192	222.30	331.12	372.52	413.91	591.20	632.86	674.53	716.19
16384	334.96	508.34	550.05	591.76	900.44	942.98	985.52	1028.07
32768	560.27	863.41	906.06	948.71	1515.75	1559.42	1603.09	1646.76
65536	1011.21	1572.29	1616.19	1660.09	2746.38	2792.30	2838.22	2884.14
131072	1912.48	2990.05	3036.46	3082.88	5220.24	5273.18	5326.11	5379.05
262144	3714.99	5830.60	5884.53	5938.46	10142.83	10204.79	10266.74	10328.69

Figure 3 – VM Overhead Table

SOURCE: FROM VMWARE.COM (HTTP://PUBS.VMWARE.COM/VSP40_I/WWHELP/WWHIMPL/COMMON/HTML/WWHELP.HTM#HREF=RESMGMT/R_OVERHEAD_MEMORY_ON_VIRTUAL_MACHINES.HTML#1_7_9_9_10_1&SINGLE=TRU)

Virtualization administrators can think of memory as a "timeshare" where it is provided to each virtual machine, as needed, and freed up when available to be used by another virtual machine, also as needed.

VMware employs a variety of features that make memory over-commitment possible. Without such features like memory ballooning, transparent page sharing (TPS), and compression, an ESX/ESXi server would simply run out of memory and crash.

Memory Ballooning

When an administrator installs VMware Tools, the memctl driver (a.k.a. ballooning driver) is installed in the guest OS. This driver creates a bubble or “balloon” of memory consumed inside the guest so the OS sees it as being used by an application. The hypervisor then takes the physical RAM freed up by inflating this balloon and allocates it to other VMs that require it.

Memory ballooning introduces a small amount of processing overhead, and if it forces a guest OS to begin paging to disk, this can significantly slow down the application(s) on the VM. If the VM isn’t using the memory, then ballooning itself isn’t a serious performance issue, but it is an indication that physical memory on the host is becoming scarce.

One of the biggest advantages of memory ballooning over other methods for handling memory over-commitment is that the memory ballooning driver allows the guest operating system to choose which pages are relinquished to the hypervisor for other VMs. This way, pages which aren’t in active use can be safely freed up, causing almost no performance impact from the guest’s perspective.

Transparent Page Sharing

Transparent page sharing is the “de-duplication” of memory that permits identical virtual memory pages to be collapsed into a single page within the host’s physical RAM, thereby freeing up memory for other uses. For example, if multiple virtual machines on a host are all running the same operating system and application, the hypervisor will compare pages of memory through hashing to locate identical pages that can be freed up through their consolidation. Ballooning and transparent page sharing work together to ensure that over-committed memory doesn't cause performance issues for the applications in the guest virtual machines.

Memory Compression

Memory compression and disk swapping by the hypervisor are the last-ditch efforts by ESX/ESXi to keep the hypervisor from crashing when memory resources on the host are stretched to the breaking point. The compression of memory pages by the hypervisor also causes additional processing overhead; however, this overhead is small in comparison to the slow-down caused by swapping pages out to a storage device. Users of vSphere 4.1 and above will be able to take advantage of this feature to reduce the amount of swapping taking place when physical memory resources are close to being exhausted.

VMware Performance Metrics to use for VM Memory Usage Analysis

Virtualization administrators should take the time to become familiar with critical vSphere counters that are used to analyze memory performance. These counters can help prevent VM memory performance issues, as well as help resolve them when they occur.

The two most commonly used vSphere memory metrics are memory active and memory consumed. Normally, when viewing memory usage on a VM within the VI client, the memory active metric is displayed (see Figures 4 and 5). The memory consumed metric is displayed by default at the host level.

- **Average Memory Active in KB** (memory.active.average) – This is the amount of memory that the guest VM is actively using, and is estimated based on recently touched pages (the smallest of active, consumed, and granted). Monitor this metric for a high memory active threshold that is approaching total memory capacity on the host. This metric doesn’t show the full “footprint” of a VM in memory, so it can’t be used reliably for determining a VM’s minimum memory requirements. However, it does give visibility into how much memory paging activity is taking place, and so can be used to judge whether the applications on a VM are engaged or idle.
- **Average Memory Consumed in KB** (memory.consumed.average) – This is the amount of guest physical memory consumed by the virtual machine for guest memory, and is calculated as the amount of memory granted (configured for the VM) less the amount of memory saved by memory sharing techniques. This metric has classically been the better choice for determining what a virtual machine’s total memory requirements are; however, there are additional complexities that can make it less helpful on modern installations. Factors such as large memory pages, transparent page sharing, and caching activity within the guest can cause this metric to become less informative.

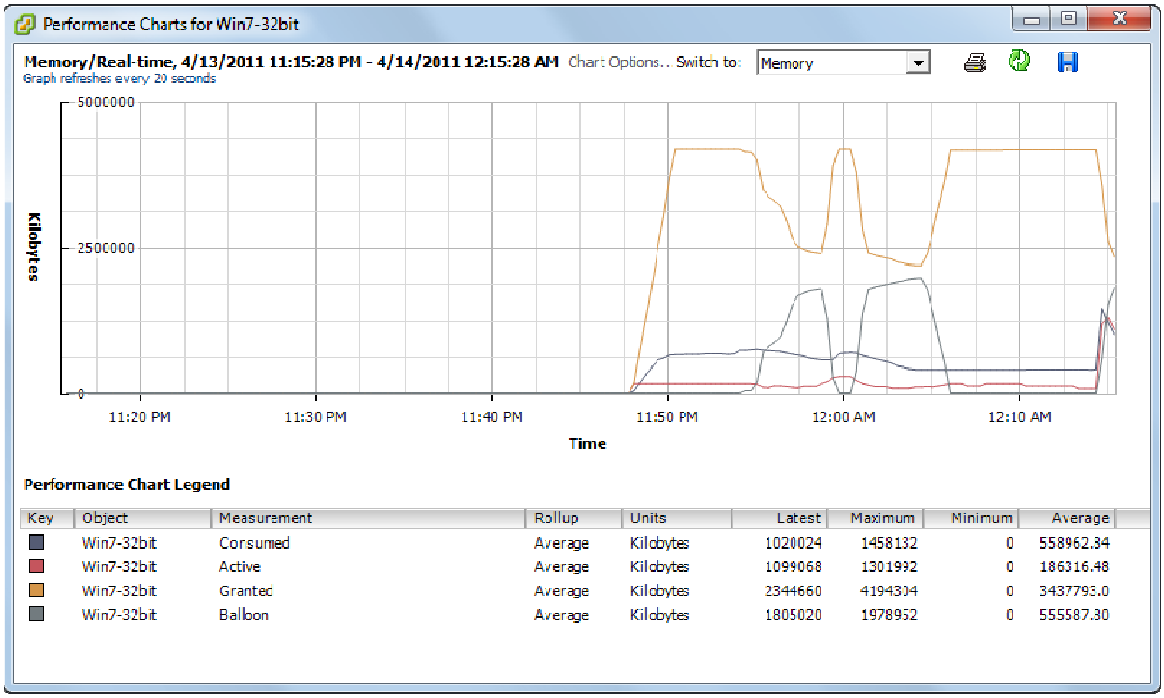

Figure 4 – A Memory Utilization Chart from within VMware vCenter

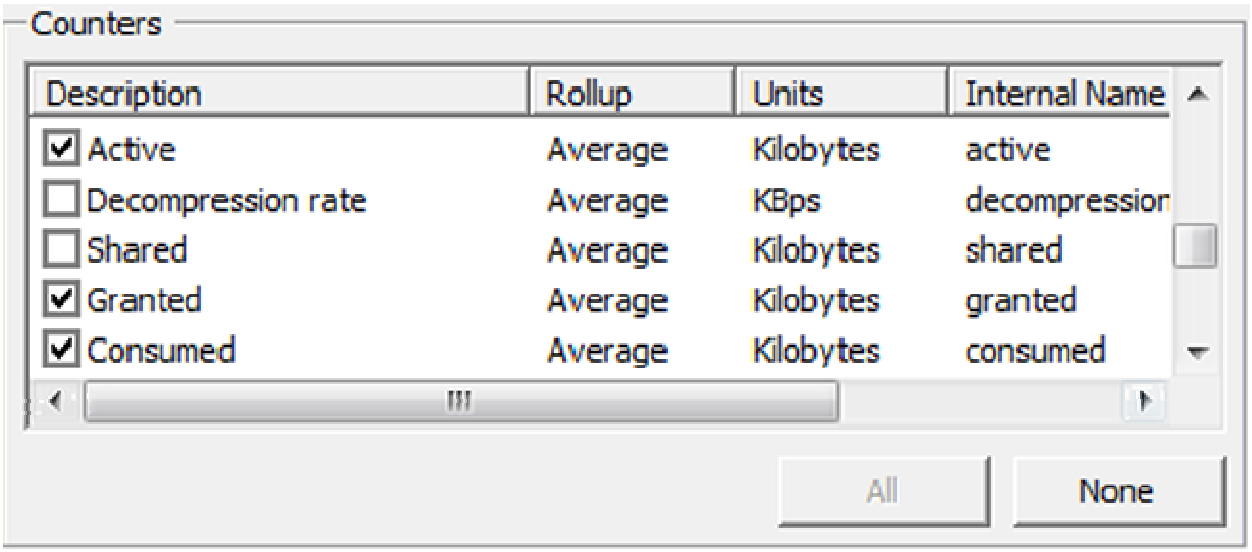

Figure 5 – vSphere Counters for Measuring Memory Performance

While memory active and memory consumed are typically the two most telling VM memory metrics, there are many more that can be used to learn about memory consumption, allocation, and sharing. More information on VMware memory metrics can be found here: http://www.vmware.com/support/developer/vc-sdk/visdk400pubs/ReferenceGuide/memory_counters.html

For example, when a host or virtual machine is swapping out to disk, this is an immediate indicator of a memory problem. Statistics like Average Memory Swapped in KB (memory.swapped.average) and Average Memory Reclaimed by Ballooning (mem.vmmemctl.average) show that a VM is having to swap or employ ballooning to get the memory that is needed.

Even when a virtual machine guest OS is no longer connected to physical hardware, it can still be useful in troubleshooting performance problems. The guest can determine what applications are running and how much memory those applications are using. If those applications are business-critical and need more memory, the VM memory should be resized. Also, it is essential to have VMware Tools installed so that the host has the ability to employ ballooning if there is any memory contention.

Knowing which metrics to evaluate is not enough to determine adequate memory sizing. To prevent and solve vSphere memory performance issues, it is important to note the thresholds for those metrics, which when crossed, are indicative of VM performance issues. Here are four such thresholds that are important to know:

- High host memory consumption > 80% (memory.active.average)
- High VM memory consumption > 75-90% (inside the guest OS)
- Host or VM Swapping (on the host, memory.swapped.average) > 0
- VM Ballooning (mem.vmmemctl.average) > 0

Another factor to consider is the use of large memory pages. Large memory pages increase the performance of memory translations for applications that use large amounts of memory. Large memory pages are enabled by default in ESX/ESXi, but must also be enabled in the guest operating system. For more information on the performance of large pages with vSphere, consult VMware's Large Page Performance Study. For more information on vSphere Memory Resource Management see VMware: Understanding Memory Resource Management in VMware ESX Server.

Determining if a VM has Memory-related Performance Problems

If a VM is already experiencing a performance problem related to memory, then the available RAM is insufficient whether due to virtual memory assigned to the VM or the physical memory installed on the host. Indicators of an impending host or cluster-wide memory shortage include when active memory approaches total memory and/or when swapping is occurring frequently. If host memory is the problem, either there is not enough or the distributed resource scheduler (DRS) is not being used properly to balance the memory load across other hosts.

In addition to the physical host memory shortage that many system administrators are familiar with on a traditional server (not using virtualization), there are other ways that memory shortages can occur in a virtualized environment.

VM memory shortages can occur if resource controls are configured incorrectly (perhaps a reservation or limit has been set inadequately) or too many memory-intensive virtual machines have been deployed

on a single server. To identify a virtual machine memory shortage, administrators should analyze memory resource settings (reservations and limits, which are covered below), as well as overall VM memory demands.

Another possibility is that VMs have been grossly over-committed and memory overhead associated with those over-commitments is consuming too much memory resources. The amount of memory being used for managing memory, per ESX/ESXi host, is available by monitoring the mem.overhead.average counter.

Finally, application-level memory shortages can occur inside the guest OS if there is a memory leak in an application or there are simply too many applications demanding memory for the amount of virtual memory allocated. To identify an application-level memory shortage, administrators can look inside the guest OS, just as they would on a traditional physical server, starting with Windows Task Manager.

Additional Factors to Keep in Mind when Sizing VM Memory

"Memory Hog" Applications

Certain applications are designed to be "memory hogs". No, this is not a reference to malicious applications. Instead, the reference is to applications that may be quite familiar to most data centers, such as an Exchange Server, Oracle DB, or a SQL database. These applications (and others) are designed to reserve (cache) all available memory when they start. While this works fine when the application is running on a physical server, it is the most inefficient way to manage memory when the application is running as a virtual machine. Check with the application vendor to determine what the true memory needs are for the deployment as a VM, and supplement that information with a look at the memory active metric within the VM to determine just how much of the memory is being actively accessed.

Limits

By configuring a limit on a virtual machine, a cap is placed on the maximum amount of memory that the VM can use. Without a limit, the allocated memory is the limit for the virtual machine. Limits are enforced by VMware Tools through memory ballooning, which keeps the VM from utilizing memory above the limit setting.

VM memory limits are often put in place by administrator who may not have fully understood the poor performance and wasted resources that limits can create. When looking at memory utilization, these memory limits skew the memory metrics and can cause confusion while troubleshooting. As general practice, memory limits are not recommended by VMware except in rare cases of serious physical memory contention.

Reservations

Memory reservations set the minimum amount of memory that a virtual machine (or resource pool) can access. Even though that memory may not be needed by the VM at all times, a reservation, in effect,

pulls that memory away from all other virtual machines that may need it. Like limits, reservations are better set on resource pools rather than on individual virtual machines, as they can affect the performance of other virtual machines and skew memory metrics.

Knowing an Application's Needs

Taking the time to look inside the virtual machine to analyze the memory that an application uses can yield a great deal of information as to that VM's memory needs. When evaluating a Windows operating system, Resource Monitor and Performance Monitor can be run to expose which processes use memory and how much memory they use. In addition, consulting the application's business owner helps determine other pertinent information, including when this application is used, who uses it, what would happen if it were unavailable, and how the use of the application is growing.

By taking time to understand these factors, VM administrators can draw further insights into properly sizing memory for the VM, configuring resource controls (if needed), and understanding the priority or relative importance of the application compared to others.

Instituting a VM Memory Sizing Process

Not only is a virtual environment itself dynamic, the usage of the individual applications in the VMs will also be in a constant state of flux. As a VMware administrator working on a critical production virtual infrastructure with hundreds or thousands of constantly-changing virtual machines, there must be a formal process for memory sizing other than just a "rule of thumb" or "trial-and-error". This process, ideally, will involve the application administrators and will be undertaken both when a new VM is being created for an application and on a periodic basis.

The workflow diagram below introduces a best practice for how to execute such a process. While this process may require slight modification in certain situations, it will work as-is for most VMware administrators. The next sections follow the initial "No" path (not a newly deployed VM) in the process step-by-step, followed by coverage of the "Yes" path.

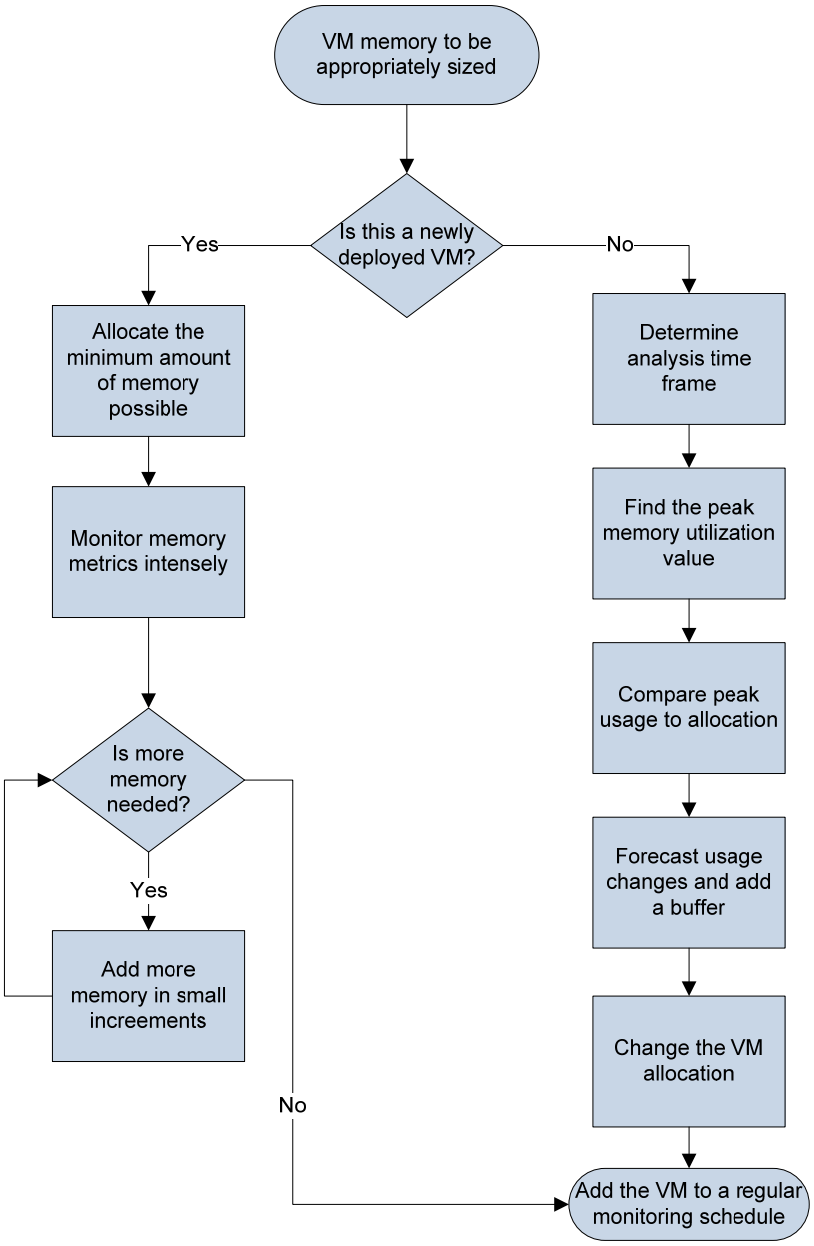

Figure 6 – The VM Memory Sizing Workflow

Determining Time Frame for Analysis

It is rather obvious that memory sizing for a VM must be done when the virtual machine is first created, but what about at other times throughout its use? Wasting expensive server resources, especially memory, on oversized virtual machines decreases ROI and may accelerate the need to purchase additional hardware. On the other hand, slow-performing applications caused by undersized VMs will lead to end user complaints and can affect a business' bottom line. Additionally, application usage can vary dramatically over the course of a day, week or month, making it necessary to allocate for the peak.

Frequent memory sizing is, therefore, necessary beyond an annual review, or worse, leaving a memory allocation as-is from the time of a VM's creation!

Here are three tips for periodic memory sizing:

- **Use Recent Data** - Make sure that the most recent memory statistics are being used, as last year's (or even last month's) usage may no longer be valid.
- **Choose a Time Frame That Captures Peak Usage** - Make sure that the VM memory configuration value selected captures the time of peak usage that will occur during a designated time period for that application. For example, use a day for a virus scanner that performs a daily scan, and a month for an accounting application that closes the books every month.
- **Control for Seasonality** - For some organizations, such as universities or retailers, seasonality must be taken into account as demand for resources will peak during student registration or during the holiday shopping season, respectively.

Finding the Peak Value

This is one of the most challenging aspects of the analysis. Determining a VM's peak memory usage cannot be reliably accomplished 100% of the time using VMware's metrics alone. The memory active metric is an estimated value, and even when measured at a peak, only indicates the level of page activity. It is, therefore, a significant under-estimation of what a virtual machine's memory needs may actually be. The memory consumed metric, by contrast, could potentially be more helpful, but in most modern virtual deployments it often produces the same value as the memory granted metric, resulting in little or no helpful insight. Whether due to large memory pages skewing these metrics, or a greedy guest OS and/or application(s), memory utilization often rises to above 95% nearly all of the time, and increasing a VM's memory allocation will not decrease the percentage utilization.

So what peak value does a virtualization administrator choose? This is an on-going issue for VM administrators everywhere, and the answer is never the same in all cases. Looking inside the guest OS can potentially give deeper insight, but only if individual processes are monitored, and only if the OS delivers metrics that distinguish between wired, active, free, swap, etc. Since presently there is no easy way to do this with the metrics available in vCenter alone, performing this assessment at scale can become quite challenging. Until more meaningful metrics are available, or VMware Tools gain deeper OS memory information, determining the peak memory usage per-VM will likely continue to be more of an art than a science.

For the time being, a good practice is to examine peak memory consumed. If that value seems abnormally high and grows equally with the allocation of additional memory, then check within the OS for any active disk swapping (independently of ESX swapping). If none exists, the memory consumed metric is not very informative, and OS-level analysis will be required.

Comparing the Peak Usage Value to Allocated Memory

The true peak value observed should then be compared to the memory configured for the virtual machine. The process should to be:

- Take a percent of the memory usage at the peak vs. allocated
- If less than 75%, then changes likely need to be made
- If greater than 75%, then start evaluating application growth

For example, if the VM is configured for 8GB and, based on analysis, the typical memory active for that VM is around 4GB (even taking long term peaks into account), then 50% memory utilization of the allocated memory is the peak value. With that ratio, it is likely that changes need to be made to the configuration. However, additional analysis may be needed to determine exactly what the allocation should be as detailed in the steps below.

Forecasting Usage Changes until Next Memory VM Allocation Review

Before deciding to downsize a VM's memory configuration, it is important to consider the fact that the memory demands of the VM (and, more specifically, its application) could increase between now and the next time the memory sizing process is conducted.

Has the application owner been contacted? Was an increasing trend in memory demand noted based on historical analysis? It is important to ensure that sufficient memory is provisioned for both current usage and estimated usage growth, or a VM could face memory-related performance issues.

Factor in a Buffer Value

Besides factoring in an expected growth rate, adding a buffer is important to ensure that the memory allocation is sufficient. While historical peaks have likely been taken into account, it is still a good practice to factor in a buffer to ensure that the virtual machine's memory has some headroom to avoid problems. For less critical applications, a buffer may not be necessary, and can be avoided to preserve more memory for more important applications. For the more critical applications that do have dynamic memory requirements, adding a 25% additional buffer on the amount of memory allocated for the VM is recommended. This way, in the event that an unexpected, business-critical demand is placed on memory, the additional memory is available to fulfill that need.

It is important to keep in mind that certain VMware features, such as transparent page sharing and ballooning, help to optimize memory usage within applications and claim memory when absolutely necessary. Thus, there is some leeway in determining what the allocation should be to balance between not wasting memory and not under-provisioning a critical VM.

Change the VM Sizing Allocation and Document the Change

Now that the timeframe for analysis has been determined, peak utilization has been determined and compared to the allocated amount, and an estimate of future memory usage has been made, it is time to take action: adjust or “rightsize” the virtual machine's memory sizing.

Unless the correct guest OS is installed and memory hot add is enabled (assuming the memory allocation needs to be increased), the guest OS must be shut down to add or remove memory from a virtual machine.

Once the guest OS is shut down, a virtual machine's memory can be resized by right-clicking on the VM (in the hosts and clusters inventory) and clicking on **Properties**. At that point, the window below appears (see Figure 7) where memory can be resized on the **Hardware** tab.

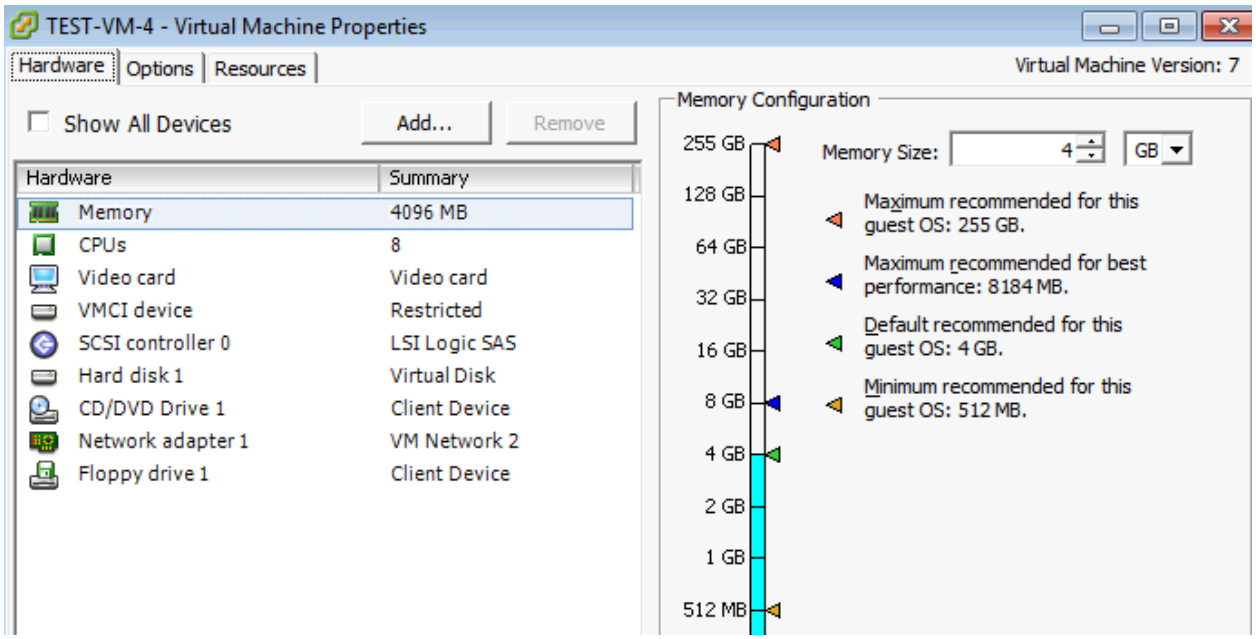

Figure 7 - Configuring Memory on a vSphere VM

Once the VM's memory is resized, and the VM is powered back on, this change should be documented (always a good practice!).

Setting Up a Regular Memory Review Based on an Appropriate Timeframe

Because application usage changes, memory allocations must be continuously evaluated to ensure that performance will not be impacted due to additional changes in the dynamic virtualized infrastructure. A regular memory review process should, therefore be set at appropriate intervals. The recommendation here is that this process occur at least once per month; environments that are growing quickly may want to do this more frequently. Once a “rightsizing” process has been conducted a few times on the virtual machines in any environment, the steps will become more familiar and administrators can then build on this experience to streamline the process.

Provisioning a New VM for Appropriate Memory Allocations

This section begins the step-by-step coverage of the “Yes” path (provisioning a newly deployed VM) in the memory sizing workflow in Figure 6. There may not be any historical performance data for a new virtual machine, unless the application is very similar to an existing VM. For this reason, the memory configuration made for a new VM is likely to be much less accurate, and the VM will need to be monitored much more closely. Still, following the sizing procedure outlined below with frequent monitoring immediately after the VM is created will provide the necessary insights to size the memory of that VM as accurately as possible.

Initially Allocate the Minimum Possible

In many cases, there will be some idea of what the memory allocation for a new VM should be. Perhaps other similar VMs exist in the environment, or another administrator who has run that application in a virtual infrastructure can provide lessons learned. Even if this information is both available and dependable, the process of monitoring and reviewing described below should still be followed.

When creating a brand new VM for an application that has not been previously deployed, and there is no indication about what the memory allocation should be, vSphere will make a memory recommendation when the VM is created, based on the guest OS specified. This is available in Figure 7 above, where it says "Minimum recommended for this guest OS is: 512MB". Here, in the absence of information to the contrary, an administrator may as well accept take the default memory configuration for the guest OS.

While it may be tempting to go with a "gut feel" or use the same amount of RAM that physical servers are configured with (e.g. just use 4GB or 8GB), it's likely that the allocation may be over-allocating the amount of memory actually required owing to the vSphere memory conservation techniques mentioned above.

Here are three additional reasons for provisioning new VMs with a smaller memory allocation initially:

- **It is technically easier to add memory than to take it away** - With certain guest operating systems, if memory hot add is enabled, RAM can be “hot added” (but there is never an option to "hot remove” RAM).
- **The OS and hypervisor will often provide different memory usage data** – The OS can only report on its usage of the memory that the hypervisor is providing. This means that the memory usage rates will often be different for the hypervisor and the OS. These differences make it difficult to pinpoint exactly how much memory is needed, however, and can cause application owners to become overly concerned there are insufficient resources to maintain performance. By slowly adding memory after beginning with a minimal allocation, it will be easier to assess the impacts of changes to reach the optimal amount.
- **Application owner concerns, Part II** – Application owners may become uneasy if memory is being taken away from an application, thereby making it more difficult to remove the memory, even if the data show that the allocation is currently over-provisioned.

Monitor Memory Metrics Intensely

Once a new VM’s memory has been allocated, that memory usage should be monitored intensely. Just how frequent and in-depth this monitoring should be depends on the criticality of the VM. Administrators may also want to use vSphere alarms to alert them as to if and when a new VM is running low on memory. When monitoring, look both at the vSphere VM memory level and inside the guest OS to see how much memory is being utilized. Is the VM maxing out the minimum amount of memory configured? If so, then more memory is needed.

Add More Memory in Small Increments

If the new VM is running low on memory, more memory should be added in small increments. Remember, memory is normally the most constrained resource a server has. Adding one or more GB of RAM on a VM may be wasteful and unnecessary. When more memory is needed, it is better to use increments of 256MB or 512MB.

Add the New VM to the Regular Memory Sizing Review Process

Every new VM should be included in the periodic memory review process (discussed above) after the memory usage has been initially stabilized with an appropriate memory allocation.

Conclusion

Memory is typically the most constrained computing resource in a virtualized data center. By leveraging the information presented in this whitepaper and implementing the workflow detailed above, virtualization administrators can more efficiently use this precious (and costly) computing resource. Rightsizing memory allocations makes more memory available for other VMs, resulting in increased VM density and deferred purchases of hardware. Additionally, regular monitoring of memory usage enables VM administrators to proactively detect problem areas in VMs that are under-provisioned to avoid performance issues. Ultimately, accurately sizing memory in virtual machines results in a better return on investment for any virtualization initiative, and application owners have greater confidence in the performance that virtualization provides their applications.

About the Author

David Davis is the author of the best-selling VMware vSphere video training library from Train Signal. He has written hundreds of virtualization articles on the Web, is a vExpert, VCP, VCAP-DCA, and CCIE #9369 with more than 18 years of enterprise IT experience. His personal Website is VMwareVideos.com.

About the Sponsor

VKernel is the number one provider of performance and capacity management products for virtualized data centers and cloud environments. The company's award-winning, easy to use and powerful products simplify the complex and critical tasks of real-time VMware performance monitoring, capacity planning, resource optimization, reporting and chargeback for virtual environments. Used by over 50,000 virtualization administrators, VKernel's products have proven their ability to immediately identify and resolve VM performance problems, maximize capacity utilization, and reduce virtualization costs.