VMware Performance Monitoring to Avoid Slow VMs

Find avoidable performance issues by monitoring vital resource areas

WHITEPAPER BY ALEX ROSEMBLAT

Table of Contents

Proactive Monitoring Helps Maximize ROI and Maintain High VM Performance

In a virtualized environment, applications compete for shared, dynamic infrastructure resources such as storage, CPU and memory. This resource sharing makes capacity management in a virtual environment an order of magnitude more complex than in a purely physical environment. As a result, proactive capacity management is critical as once problems manifest themselves, they can be difficult to diagnose.

Why is diagnosing capacity management problems difficult? First, aside from monitoring resource management for each VM, VMs interact with one another and resource usage must be balanced at the host, cluster, and resource pool level. This interconnectedness of the virtualized environment means that a problem with a VM or a host may not be directly attributable to that virtual object, but may instead be caused by a completely different VM experiencing problems, or even bigger issues at the host, resource pool or cluster level. This makes troubleshooting performance problems time-consuming and complicated. Issue investigation in a virtualized environment requires analyzing multiple resources' data from several infrastructure areas to find the cause of an issue. Proactive capacity management can help provide insights into the complicated nature of the virtual environment and yield significant value by:

- Discovering which resources are causing existing performance problems
- Recognizing warning signs of impending performance issues
- Uncovering performance issues that had not yet been detected in the environment
- Detecting if capacity problems occur at certain time periods throughout the day
- Calculating exactly how much capacity is available for new VMs
- Measuring if VMs are overallocated and wasting hardware resources
- Identifying data objects that are no longer used, have been forgotten, and are wasting hardware resources
- Gauging resource usage growth rates to forecast when new hardware will be necessary
- Finding areas with capacity availability to move VMs to and better balance a virtual environment

This whitepaper identifies four VM resource areas that can be proactively monitored, and will describe how to conduct that process.

The Four VM Resource Areas to Monitor

Although many components affect data center ROI and VM performance, four particular resource areas can be proactively monitored with a small amount of effort to great effect. These areas are: CPU Utilization, Memory Utilization, Storage Utilization, and Disk I/O Latency. The data required to engage in this monitoring can be extracted from vCenter and then analyzed. Below are explanations on what data to extract and what to do with it to spot issues.

CPU Utilization

The CPU of a computer processes commands. If CPU resources have been spread thin with too few physical CPUs available to support VM needs, these physical CPUs become overutilized in two ways:

- First, a CPU will attempt to share its processing bandwidth with several processes at one time. It will work part of the way through one process, and move to the next, continuing to work on the partially completed processes in an incremental fashion until a process is fully completed, and then the CPU will accept another process and continue on. However, a CPU has limited processing bandwidth, and when it hits its capacity, processes will have to wait their turn before being addressed by the CPU.
- Second, if a VM has multiple CPUs allocated, processes from the VM will wait until the exact number of CPUs is available. For example, if a VM has two CPUs allocated, and only one CPU is available in the host when processing is needed, commands will not be processed until exactly two CPUs are available. That lone CPU that could have been used will remain idle, decreasing the available CPU processing capacity in an environment.

Once either or both of these conditions occur, CPU capacity will diminish and processes may need to wait in line. This wait time becomes a VM performance issue known as CPU Ready. CPU Ready will dramatically bring down VM performance to all VMs that are sharing the CPU resources. There are ways to pinpoint CPU Ready as well as areas to look into that warn that CPU Ready could occur. CPU utilization can be proactively monitored by:

- Checking for CPU Ready by monitoring all VMs' cpu.ready.summation metric for any values other than zero. This VMWare metric reports on how much CPU Ready is occurring in an environment. If there are any non-zero values, it means that a capacity bottleneck is occurring which should be resolved.
- Checking for high CPU utilization by monitoring the cpu.usagemhz.summation metric for all VMs as often as possible. This VMWare metric measures CPU usage. If there are any values that are inching close to 100%, it means that with more utilization, CPU Ready can occur. This value is the early warning indicator that CPU problems may begin to arise, that VMs need to be balanced within existing hardware, or that more hardware needs to be added.
- Assessing historical cpu.ready.summation values for patterns when CPU ready has occurred (when there are non-zero values). As different VMs experience peak utilization at different times, it is possible that CPU Ready issues only occur at certain periods. If this is the case, catching this issue can avoid recurring VM performance problems.
- Tracking where available CPU capacity exists based on low utilization values in the cpu.usagemhz.summation metric. Knowing where new VMs can be placed or where to move VMs that need additional CPU capacity can be immediate fixes to capacity bottlenecks or a buffer while additional equipment is added.
- Comparing historical CPU usage per VM to the allocations that have been made for that VM and finding the highest peak values. If there is a significant difference between the CPU usage peak

values and the CPU allocation that the VM has, there is CPU resource that has been wasted and can be reclaimed to add CPU capacity in the environment.

Memory Utilization

Memory is typically the most constricted host resource, and the most likely to begin suffering bottlenecks if proactive monitoring is not in place. There are many factors that go into memory management making this a complicated resource area to monitor for optimal performance allocation. At the time of writing, there is a lack of metrics which reveal sufficient detail to derive optimal memory allocation for a VM. Yet, proactive monitoring of existing metrics can help easily spot memory problems, or can serve as warning signals for impending problems. Memory can be proactively monitored by:

- Checking for memory swapping by monitoring the mem.swapin.average, mem.swapout.average and mem.swapped.average metrics to see if they have a value. If these metrics have non-zero values, memory swapping, a computer action that occurs when VMs do not have enough memory to work on their tasks is occurring. Swapping will dramatically bring down VM performance and could begin to affect other areas such as throughput capacity and disk performance based on the fact that large chunks of information are communicating between the server and storage area. If there is a value in these metrics, investigate immediately.
- Checking for ballooning by monitoring the mem.vmmemctl.average metric. Ballooning is when VMWare takes memory that is not being actively used away from other VMs on a host to give to a VM that needs additional memory. If any ballooning is occurring, it is a warning that memory swapping may begin to occur, as a VM that has had its memory taken away may have to swap if it begins to experience increased activity. Swapping will lead to significant memory problems.
- Adding the mem.overhead.average metric value to the memory allocations for all VMs on a host to get an accurate picture of how much memory has actually been allocated in the host. An environment will have less memory available than what may be assumed based on allocations for existing VMs. Hosts require additional memory to manage the use of memory in VMs. Thus, the summed memory allocation of all VMs drawing from a host's memory is not the actual total amount of memory allocated in the host. The overhead amount must be added in.

Storage Utilization

Storage is a key resource for a virtualized environment. After all, a VM is technically a VMDK file, which can become a large data file as an operating system and other supporting software is included. Virtualization has made provisioning servers much easier and cheaper leading to a proliferation of VMs, and thus data creation in the data center. Also with additional data objects related to VMs such as snapshots and VM templates, each VM is often the cause of several times more storage usage than just its own VMDK file. Also important to note, different VMs will grow their storage needs at different rates based on usage levels and user activities. Free storage availability is particularly important to track as without enough storage, a VM will crash.

Storage can be proactively monitored by:

- Tracking how much storage is being used by all VMs, snapshots, and templates to know exactly how much storage is available, and to be able to weather an equipment purchasing cycle with a healthy buffer.
- Tracking storage usage for all VMs to assess growth rates. These rates can then be extrapolated to forecast future storage needs.
- Investigating the last usage or modification dates on all powered-on VMs, snapshots, powered-off VMs, templates, and VMDK files. If a snapshot, VMDK file, powered on VM, or template has not been used in many months, it's likely that someone may have forgotten about that file, and that file is wasting storage space. In the case of a powered-on VM that is no longer used, server resources are also being wasted. Deleting these files will help free up storage space to add on to the free storage space total.

Disk I/O Latency Levels

A VM must move information from readily available memory to disk storage as part of its operations. Usually, these transactions involve large amounts of data moving back and forth within limited bandwidth. The process involves data being received by the disk, written or read, and then a response from the disk getting returned to indicate whether the operation was successful. Since all operations have limits, there is limited bandwidth between a host and storage and also a limited amount of data a disk can read or write at a given time. If the bandwidth becomes constricted or a disk is receiving too much usage, some operations may run into these bandwidth limitations causing an increase in latency, which is a measure of the time needed for a command to be sent, processed, and returned.

In the case of bandwidth constriction, data transactions will need to wait both when leaving the VM for the disk, and when a response from the disk must then be transmitted back to the VM. In the case of an overloaded disk, disk operations will slow as the datastore struggles to respond to all requests, and each request takes a certain amount of bandwidth and time to process. These resulting lag times are referred to as disk I/O latency and will slow down VM performance if it they occur. Disk I/O latency can be monitored so that VMs can be redirected to other datastores to balance the amount of traffic going from VM to datastore, and that the datastore must process.

Disk I/O can be proactively monitored by:

- Averaging together the disk.read.average and disk.write.average metrics to get a throughput measure which can be analyzed in different ways.
- Assessing historical throughput values at different points of the day to see if usage in VMs at certain time periods are causing throughput peaks leading to disk I/O.
- Isolating throughput outliers or high values on a VM basis. These are the top throughput-using VMs, and should be watched most carefully to see if an increase in usage in one of these could cut off throughput capacity to other VMs, causing disk I/O issues.
- Conducting trend and change analysis on throughput in all areas of the virtual infrastructure. Knowing when significant usage changes are occurring, or if throughput usage is trending

upwards can highlight areas to proactively load balance to avoid future performance bottlenecks.

- Searching for high values in the disk.totalLatency.average metric which reports on disk latency times. If a datastore is experiencing high latency, throughput for every host connected to the datastore should be examined for high values that can pinpoint what area is overloading the datastore and causing the performance problems.
- Checking for bus resets to a datastore by monitoring the disk.busResets.summation metric for a value other than zero. A bus reset is when all commands that have been queued up in an HBA or Disk Bus have been wiped out. If there is a non-zero value in this metric, investigate immediately.
- Checking for commands aborted in a datastore by monitoring the disk.commandsAborted.summation metric for a value other than zero. This metric shows the number of times a request was sent to the datastore and the command was aborted. If there is a non-zero value in this metric, investigate immediately.

Conclusion

Proactive monitoring of a virtualized data center can assist in finding potential performance problems before they occur, getting to the root of existing problems, gauging environment growth, discovering how much capacity is available, reclaiming resources, and assessing how much hardware needs to be added to an environment. By setting up processes that evaluate this information as often as possible, problems that could turn into time-critical issues can be avoided, and the resolutions put in place before an end user realizes that an environment is experiencing capacity issues.