Detecting Available VM Slots for Capacity Planning

A metrics-based approach to calculate how many more VMs can safely fit in a virtualized environment

WHITE PAPER BY KEN LATIMER AND ALEX ROSEMBLAT

Contents

Introduction

Knowing how many more virtual machine (VM) slots are actually available within existing infrastructure is vital to capacity planning in a virtualized environment. The layers of abstraction from the physical to the virtual, as shown below, can make this assessment a mix of art and science. The fundamental challenge, therefore, involves gaining visibility through these layers of abstraction to understand how the virtual resources are actually utilizing the underlying physical resources.

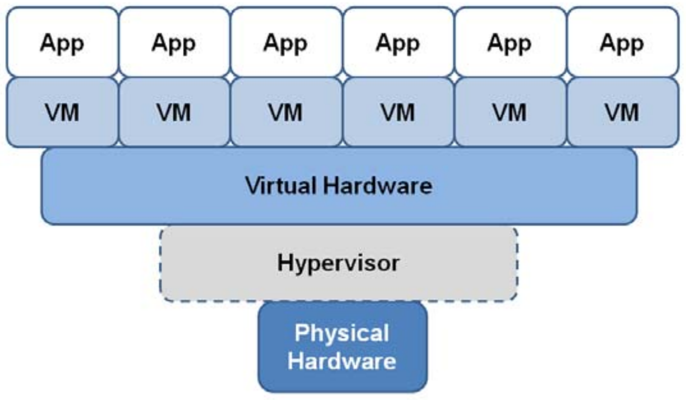


Figure 1 – Abstraction through the Layers in the Virtualization Stack Adds Complexity to Planning

Without full visibility through all layers of the virtualization stack, capacity planning requires an iterative, trial-and-error process. This is the art of capacity planning, and some IT managers are able to achieve acceptable results. But, the process is time-consuming and by definition, error-prone, and can lead to sub-optimal results.

Accurately determining available VM slot capacity becomes increasingly scientific when employing a metrics-based approach, like the one outlined in this white paper. The metrics used are readily available within VMware vCenter, and can be gathered either manually or using purpose-built capacity planning tools. Either way, the process is the same; the difference is the amount of manual effort required.

Before outlining the process, some background information is provided on the performance vs. cost tradeoff involved in virtualized environments and why Dynamic Resource Scheduling (DRS) is not a substitute for capacity planning. The metrics-based process is then introduced at a high level, followed by separate sections containing more detail on each of the steps involved. A section at the end outlines additional considerations that apply to high-availability clusters.

By reading this white paper, VM administrators will:

- Be able to extract key VM performance metrics and run calculations to derive how much available capacity remains in hardware resources
- Recognize when resource contention exists that must be addressed before engaging in capacity planning

- Be able to combine the results of assessing hardware resource availability with the resource specifications for a new VM to determine how many available VM slots remain
- Be able to factor HA into available VM slot calculations
- Gain a background on additional considerations needed to assess how many VM slots are available in an environment

The Performance vs. Cost Tradeoff

Virtualized environments afford far greater utilization of physical servers, potentially resulting in significant cost savings. In some cases, utilization rates can improve from the 10-20% range for dedicated servers, to 60-70% or more for virtual machines. In many other cases, however, the improvement in server utilization is far less owing to the over-provisioning of physical resources.

Over-provisioning is often employed as a method to guarantee good application performance, but it can undermine the cost-savings advantages of consolidation and virtualization by under-utilizing the host's resources. The tendency to over-provision is a symptom of a lack of visibility into how VMs are actually utilizing the physical resources. Looking at it another way: The only way to overcome the fear of causing performance problems by under-provisioning is to have full visibility into the virtualized infrastructure.

Without full visibility, the only way to detect over-provisioning is to monitor for performance degradation with different combinations of resource allocation. One problem with this trial-and-error process is that finding and fixing a performance bottleneck in one resource can create a new performance bottleneck in another. This iterative process must then be repeated with each and every new or changed application.

The metrics-based process outlined here enables IT managers to achieve an optimal balance between performance and cost in a more predictable fashion. In doing so, it is also important to take into account the service level and criticality of each application, as not all applications are created equal. It is worth noting, however, that an optimal allocation of resources delivers the same performance as an over-allocation. In fact, some resource over-allocations, such as allocating more CPUs than needed to a VM can degrade performance.

The Role of DRS – Dynamic Real-Time Balancing, Not Planning

The Distributed Resource Scheduler (DRS) scheduler from VMware is not a capacity planning tool, and should not be used a substitute for one. Rather, its intended use is to dynamically allocate and balance computing resources in real-time. DRS operates by continuously monitoring actual utilization across a resource pool, and then reallocating available resources among virtual machines to satisfy service levels or other objectives. DRS is also used during scheduled maintenance to migrate the VMs affected to active physical servers.

Achieving satisfactory results with DRS, however, often depends on solid capacity planning. Indeed, knowing the number of VM slots available is fundamental to the ability of DRS to take advantage of any additional capacity available. Slot availability is, therefore, an input to DRS, and not an output or result of its operation.

Assessing VM Slot Availability: The Workflow from a High Level

The diagram below provides a high-level outline of the workflow involved in assessing available VM slots in a virtualized environment, including for a high-availability (HA) cluster configuration. Such a process is required when either new applications are being added, or during a periodic capacity planning assessment when determining budgeting and procurement needs. The result, which takes into account anticipated growth for existing applications, as well as some buffer or headroom to create a safety margin, is a realistic number of VM slots available in the host server(s).

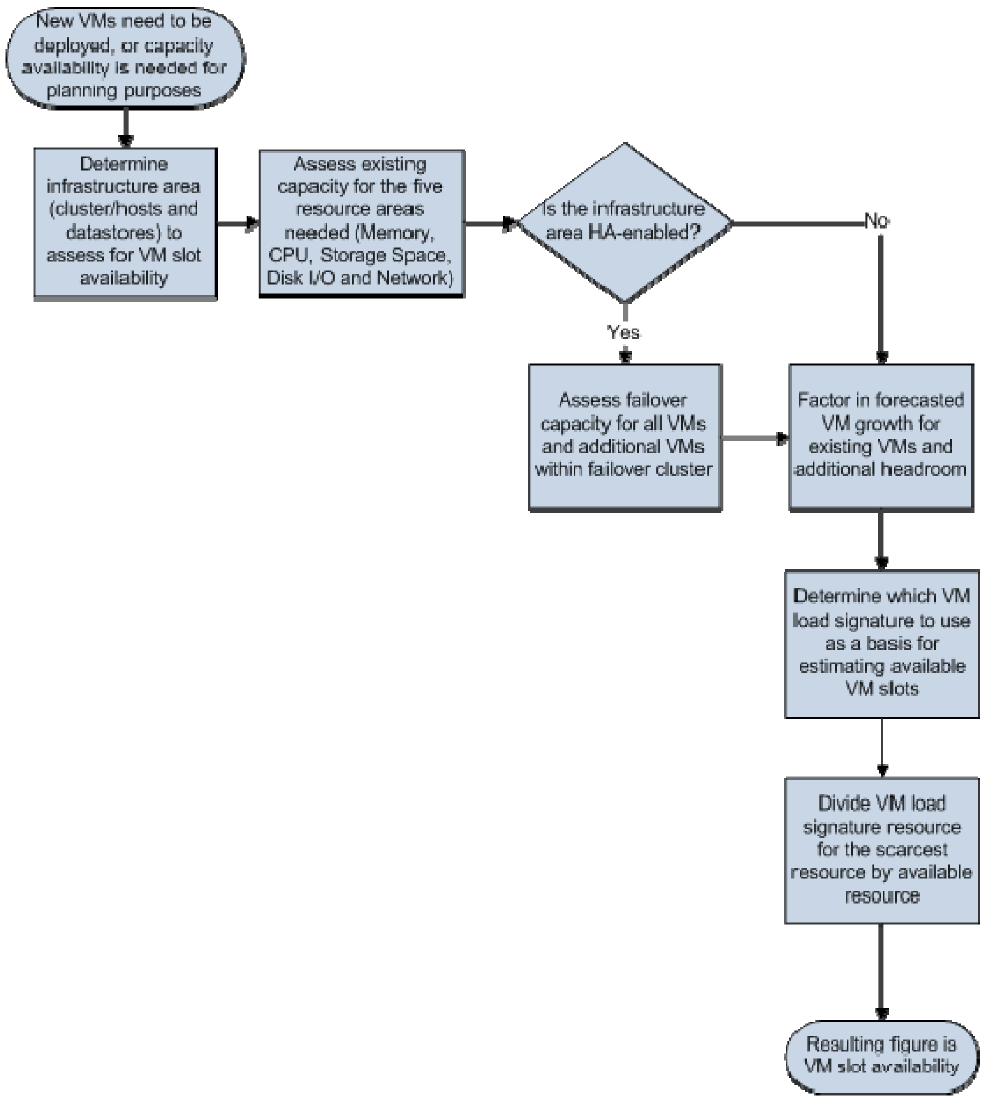

New VMs need to be deployed, or capacity availability is needed for planning purposes
Determine infrastructure area (cluster/hosts and datastores) to assess for VM slot availability
Assess existing capacity for the five resource areas needed (Memory, CPU, Storage Space, Disk I/O and Network)
Is the infrastructure area HA-enabled?
No
Yes
Assess failover capacity for all VMs and additional VMs within failover cluster
Factor in forecasted VM growth for existing VMs and additional headroom
Determine which VM load signature to use as a basis for estimating available VM slots
Divide VM load signature resource for the scarcest resource by available resource
Resulting figure is VM slot availability

Figure 2 – Overview Workflow for Assessing Available VM Slots

Before Beginning the Capacity Planning Process

The process outlined in the workflow above assumes a solid environment baseline; that is, it assumes that the virtualized environment is functioning properly with a reasonably optimal allocation of resources. This is, of course, a good if not a best practice for managing VMs. If this is not the case and an environment's resources are being used inefficiently, it is recommended that two steps be taken before beginning:

The first step is to right-size the resource allocation. This can be accomplished by trial-and-error, as mentioned above, of with a purpose-built tool which can automate the process and deliver more precise results with substantially less effort. Right-sizing minimizes the over-allocation of VM resources based on actual peak and average usage, and normally also improves VM performance by optimizing memory, CPU and storage allocations.

The second step is to eliminate any wasted resources. Such waste often exists in the form of orphaned or abandoned VM images, unused snapshots and templates, and powered-off and zombie (or idle) VMs.

Recognizing that very few virtualized environments are able to maintain a constant, optimal allocation of resources and that VMs may also be under allocated in resources, the process outlined below also includes how to identify the symptoms of performance problems that must be addressed before adding new VM workloads to an environment.

With all VMs functioning within performance specifications and the environment right-sized and all wasted resources reclaimed, it is time to determine which infrastructure areas to assess for VM slot availability.

Determining Which Infrastructure Areas to Assess

This first step in the capacity planning process involves determining which infrastructure area(s) are to be assessed in the analysis. For example, will the new applications be running in a production or a development environment? In addition, are either of these environments partitioned somehow, such as by department or function? Note that a comprehensive capacity planning effort is likely to involve all infrastructure areas.

It is also important at this stage to consider the service level requirements and criticality of each new or anticipated application, as well as any datastore considerations. These issues are particularly important in the production environment, but may also be important for performance or quality assurance testing the development environment.

Understanding the Five Critical Resources

This section is divided into five subsections that address each of the five critical resources in order of importance: memory, CPU, storage I/O, storage space and the network. During the analysis it is critical to note the use of shares, reservations and limits when evaluating resource pools. The most critical of these in capacity planning is the reservation (a minimum resource allocation), which has a default value of zero. Any VM configured with a non-zero reservation dedicates use of that particular resource exclusively to the VM, making it unavailable to all other VMs.

This section follows the “No” path in response to the question “Is the infrastructure area HA-enabled?” in the workflow diagram above. Special considerations for high-availability clusters are then covered in a separate section.

1. Memory

Folklore has it that Willie Sutton was once asked why he robbed banks, and he replied, “Because that’s where the money is.” If there is limited staff or time to do a thorough job of assessing VM slot availability, spend it on this VM resource. Having adequate memory is critical to achieving good performance in most applications, and the natural tendency to over-allocate normally makes memory the limiting factor in a virtualized environment.

The two most common symptoms of a memory contention problem are indicated in the workflow diagram below: swapping and ballooning.

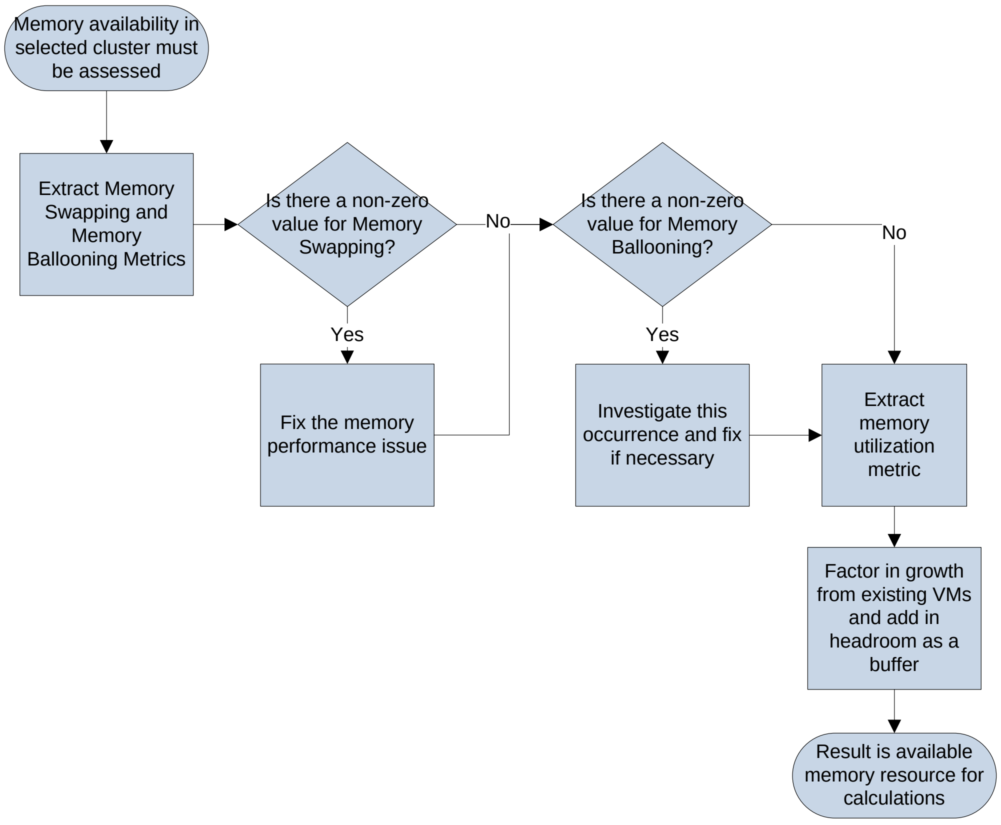

Memory availability in selected cluster must be assessed
Extract Memory Swapping and Memory Ballooning Metrics
Is there a non-zero value for Memory Swapping?
No
Yes
Fix the memory performance issue
Is there a non-zero value for Memory Ballooning?
No
Yes
Investigate this occurrence and fix if necessary
Extract memory utilization metric
Factor in growth from existing VMs and add in headroom as a buffer
Result is available memory resource for calculations

Figure 3 –Workflow for Assessing Available Memory

Assessing Contention

The occurrence of memory swapping (characterized for VMware by mem.swapin.average, mem.swapout.average and mem.swapped.average) indicates contention for memory, indicating that the VM using that memory allocation is likely experiencing performance issues. These metrics are measured at the VM level and are averaged together to yield a percentage value. All three should be at or close to zero in a properly configured environment where there is sufficient memory for all the VMs. Memory swapping also increases the amount of storage I/O required (a separate topic covered below) as data begins to transfer from server to datastore.

Memory swapping is a symptom of insufficient memory in the host or an insufficient allocation to one or more VMs, or that ballooning is occurring. To eliminate or minimize memory swapping, the memory allocation for some or all VMs will need to be increased, and further analysis will need to be conducted to determine if the swapping is being caused by ballooning.

The occurrence of memory ballooning (reported by VMware as mem.vmmemctl.average) also indicates a memory contention problem. Ballooning occurs when VM memory utilizations begin to approach a limit, either allocated or physical (the host's total memory), requiring memory to be reallocated. This metric should also be at or close to zero in a properly configured environment where there is sufficient memory for all the VMs. A value over 100% indicates that a VM has a limit preventing it from getting all the memory it needs to function properly.

As with memory swapping, ballooning is a symptom of insufficient memory in the host or, more likely, incorrect allocations to one or more of the VMs. Ballooning is a way to avoid a memory shortage for one application, but it can adversely affect the performance of other applications. And memory taken from another VM must be swapped to disk. The solution is to right-size all VMs on the host (a topic beyond the intent and scope of this white paper, but covered in more detail in VM Memory Sizing Considerations).

Assessing Resource Availability

Memory utilization for VMware is captured by two metrics: mem.active.average and mem.consumed.average. The former reports the amount of memory pages being used by a VM at a given time; the latter measures the total amount of memory being consumed by a VM. If issues are found with either metric, the usual resolution is to allocate more memory to the VM. If the current host is at or near its maximum memory utilization, the VM may need to be moved to another host with more memory and/or a lower total utilization of memory. One factor however that can skew mem.consumed.average is if the large memory pages setting is enabled. In this case mem.consumed.average may show near full utilization at all times. Gaining more precise visibility into VM memory usage in this case can become more intricate. The topic is covered in more detail in the VM Memory Sizing Considerations white paper.

Because a VM normally does not use all of its memory allocation all of the time, the mem.consumed.average metric is a more meaningful way to gauge a VM's total memory utilization. Even when large memory pages are in effect, using Memory Consumed as a baseline metric to determine how much memory is being used reflects the more conservative and performance-maintaining measure for memory usage. The total allocated memory used by VMs can be subtracted from the total memory count available in a host to determine how much memory remains available within existing hosts.

2. CPU

The CPU is next most critical resource in the assessment. The symptom of a CPU contention problem is indicated by a non-zero value for CPU Ready, as shown in the workflow diagram below.

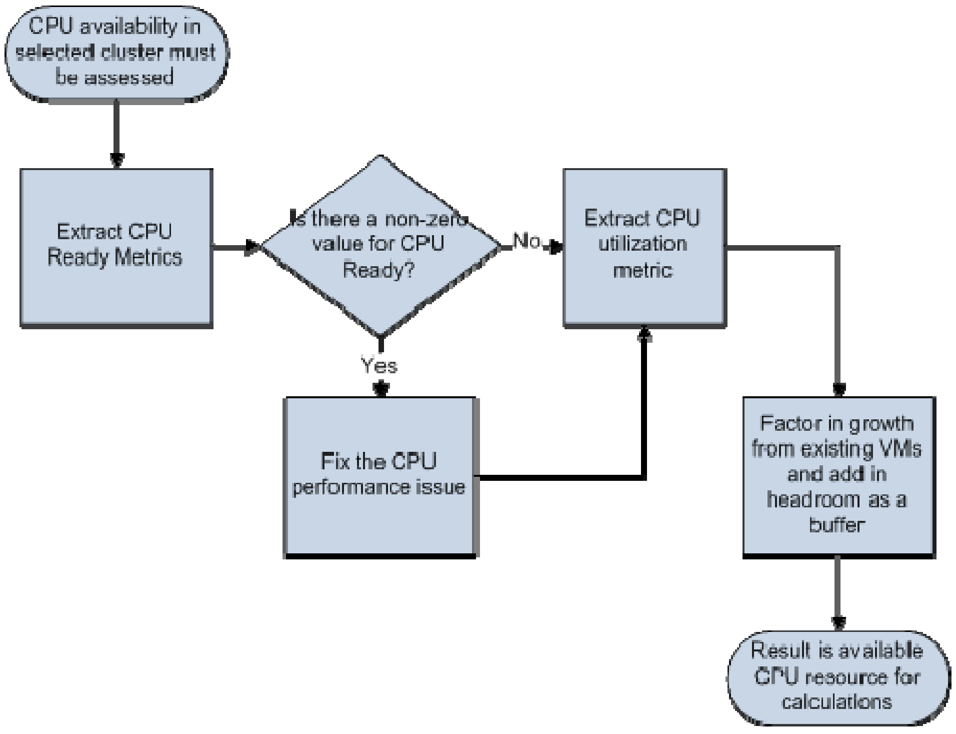

CPU availability in selected cluster must be assessed
Extract CPU Ready Metrics
Is there a non-zero value for CPU Ready?
No
Extract CPU utilization metric
Yes
Fix the CPU performance issue
Factor in growth from existing VMs and add in headroom as a buffer
Result is available CPU resource for calculations

Figure 4 –Workflow for Assessing Available CPU

Assessing Contention

A non-zero value for CPU Ready (reported by VMware as cpu.ready.summation) indicates a VM is waiting for the processor. It is a symptom of contention among VMs for processor time, and invariably has an adverse impact on performance. If this value is not zero, the performance issue should be addressed before moving on to capacity planning. The solution requires rebalancing VM loads among physical CPU cores (on the same or another host) and/or right-sizing CPU resource allocations for the VMs affected. A good guideline here, in the absence of actionable information, is to allocate a maximum of two to three virtual CPUs amongst all VMs sharing that resource to each physical CPU core.

Assessing Resource Availability

CPU utilization for VMware is captured by cpu.usagemhz.average, which reports utilization of the physical CPU at the VM level. A high value indicates that the CPU core is approaching or has reached its full capacity, making it unavailable for other VMs. It is also likely that when this situation occurs, that CPU Ready will have detected the problem and that the applications running on the VMs may be experiencing performance problems.

If CPU utilization is too high, one or more of the VMs will need to be moved to another CPU core. If the current host is at or near its maximum utilization of all cores, the VM(s) will need to be moved to another host with more CPU capacity. By extracting the total amount of CPU utilization cycles available from existing hardware specification ratings, and then subtracting the observed usage (as shown in cpu.usagemhz.average), the projected growth usage for existing VMs and any headroom needed, an exact amount of MHz available for new VMs is calculated.

3. Storage I/O

Storage I/O contention can be caused by a number of factors, and the precise cause(s) can be difficult to find. As shown in the workflow diagram below, the situation depends primarily on whether the storage is on the server or on the network. For network-attached storage, such as Fibre Channel or iSCSI, see the subsection below on the network. This section addresses storage residing on the physical server, where the symptoms of any problems include aborted commands, bus resets and latency, as shown in the workflow diagram below.

It is important to note, of course, that many applications access storage both on the host server and via the network. For example, the application software may reside on the server, while the database(s) being accessed are located on another server, or in the storage area network.

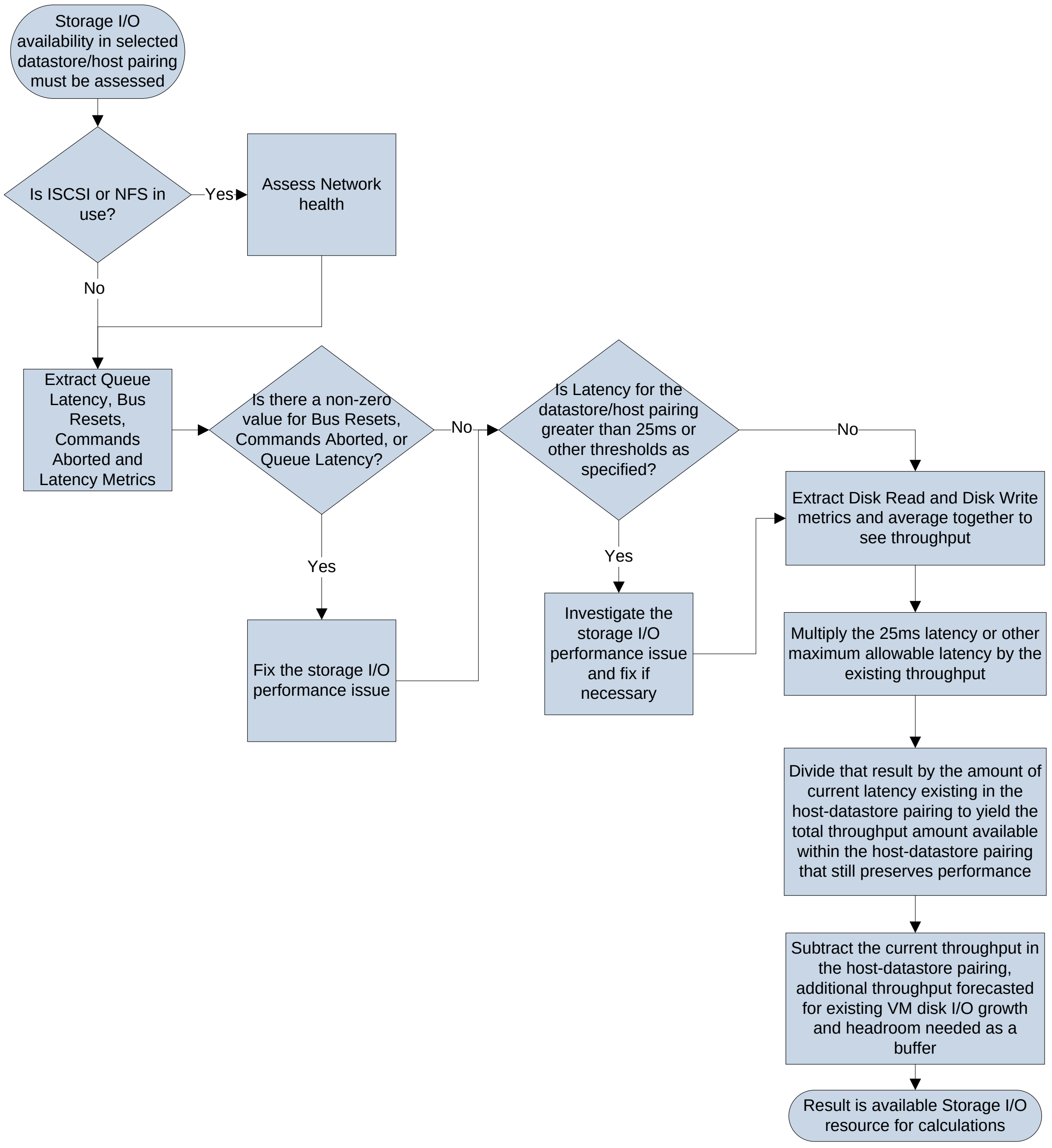

Storage I/O availability in selected datastore/host pairing must be assessed
Is ISCSI or NFS in use?
Yes
Assess Network health
No
Extract Queue Latency, Bus Resets, Commands Aborted and Latency Metrics
Is there a non-zero value for Bus Resets, Commands Aborted, or Queue Latency?
No
Yes
Fix the storage I/O performance issue
Is Latency for the datastore/host pairing greater than 25ms or other thresholds as specified?
No
Yes
Investigate the storage I/O performance issue and fix if necessary
Extract Disk Read and Disk Write metrics and average together to see throughput
Multiply the 25ms latency or other maximum allowable latency by the existing throughput
Divide that result by the amount of current latency existing in the host-datastore pairing to yield the total throughput amount available within the host-datastore pairing that still preserves performance
Subtract the current throughput in the host-datastore pairing, additional throughput forecasted for existing VM disk I/O growth and headroom needed as a buffer
Result is available Storage I/O resource for calculations

Figure 5 –Workflow for Assessing Storage I/O Availability

Assessing Contention

Problems with storage I/O are caused by contention, and normally begin with an increase in latency in the host-datastore pairing. As the problem worsens, commands begin to be aborted, normally for a single request at first, and perhaps eventually for all requests in the queue if the situation is not addressed.

Serious issues with storage I/O are captured by VMware in two metrics: disk.commandsAborted.summation and disk.busResets.summation. Values for both metrics should be zero; any positive value indicates a problem is occurring. When storage I/O commands are aborted, the application may hang or crash, depending on how well the software is able to monitor and recover from such error conditions. When the storage I/O queue itself becomes overwhelmed, the bus is reset, causing all active commands to be aborted.

These are serious problems that should be addressed urgently. The cause might be a malfunction of the disk drive or I/O interface, or it could be an anomalous result of too many concurrent requests. If the hardware checks out okay, the recommendation is to rebalance VM traffic by moving one or more VMs to another datastore or host.

Latency in a VMware environment is measured by two metrics: disk.queueLatency.average and disk.totalLatency.average. The former reports the amount of time a command waits in a queue to be processed by the disk; the latter measures the total latency for the physical disk itself. It is common for both of these metrics to increase in unison as the commands in the queue must wait longer as the disk becomes more fully utilized, thus increasing the total latency. Latency should optimally be kept below 25ms. If the latency values begin to exceed this threshold, the storage I/O load on that datastore should be immediately analyzed to head off any performance issues. It is important to note that higher disk latency could be caused by a memory problem if pages are being swapped.

Assessing Resource Availability

The next step is to evaluate the actual disk I/O (reads and writes) and compare these to the maximum capability of the datastore. Both the peak and the average rates should be noted to determine if there is any contention occurring and to assess overall resource utilization. Measuring storage I/O is difficult to do in absolute terms. Many variables such as the physical bandwidth of the various hardware pieces in the connection, the size of data packets, the frequency of data packet transfer, and the pattern of read and writes by the various VM's generating I/O load can affect the capacity available on average and peak times. However, assessing performance impacts, and thus, determining how much available capacity is available within the host-datastore pairing can be measured in relative terms:

By using command latency as a measure of storage I/O performance, the values from host to datastore pairing for disk.totalLatency.average can be used to find how much more storage I/O bandwidth exists. Assuming that 25ms is the worst storage-related performance that could be tolerated (this value may be lower or higher based on the criticality of the applications running on the host), the 25 ms can be multiplied by the existing disk throughput (found by averaging disk.read.average and disk.write.average together) in that host-datastore pairing, and then that number can be divided by disk.totalLatency.average to get a result for the maximum amount of storage I/O available while maintaining adequate performance in the host-datastore paiting. Subtracting the amount of throughput used already, headroom needed, and any growth expected for existing VMs will yield a value of how much throughput remains for new VMs to use.

4. Storage Space

While depleting all storage space on the host server can be catastrophic, storage capacity rarely places a constraint on the number of VM slots available for two reasons. The first is that disk drives are now quite inexpensive, making it possible to configure the host with copious amounts of storage capacity. The second reason is that it is very easy to detect an impending problem, and remedy the situation by deleting files and datastores that are no longer being used, or relocating a large application to another physical host with ample capacity.

There are two caveats to consider here, however. The first is the importance during the analysis to factor in growth by existing VMs to account for thin provisioning. A common way to do this is to compare over-commitment levels to existing utilization, and add sufficient headroom as a buffer. The second caveat involves the need to locate specific datastores based on the performance tier or other criteria. For example, if it becomes necessary to relocate an application, it should be moved only to a host in the same tier.

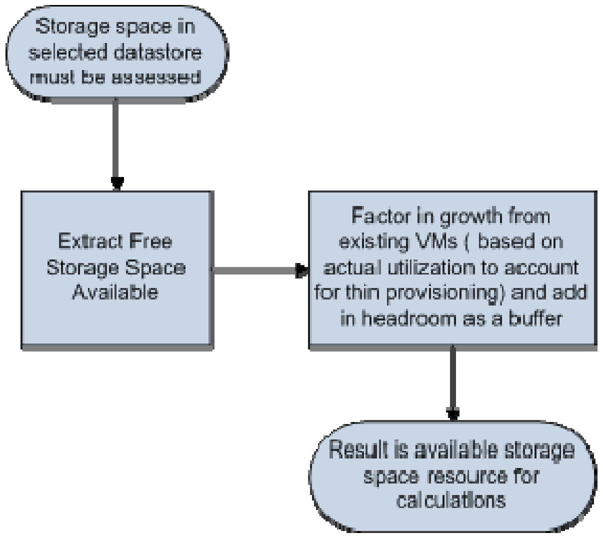

Storage space in selected datastore must be assessed
Extract Free Storage Space Available
Factor in growth from existing VMs (based on actual utilization to account for thin provisioning) and add in headroom as a buffer
Result is available storage space resource for calculations

Figure 6 –Workflow for Assessing Storage Space Availability

Assessing Resource Availability

By calculating the amount of storage space available for that VM within the datastore areas that have been given visibility to VMware vCenter, and then subtracting the amount that has been consumed already, any forecasted consumption growth, and factoring in headroom for a buffer, a VM administrator can determine how much storage space is available for new VMs. Importantly, only portions of a datastore may have been given access to vCenter to use for virtual machines. If additional storage that is not recorded in vCenter is available, those amounts may also need to be factored in to determine how much free storage space is available.

5. Network

In a well-run data center, the network should not impose any limitations on the number of VM slots supported by the virtualized servers. The network can be a limiting factor, of course, in a high-performance computing (HPC) environment. But an HPC cluster generally operates in isolation from other, more traditional applications, whether virtualized or not.

Assessing contention or other bottlenecks that can cause performance problems is further complicated by the fact that the servers, the network and the storage area network (SAN) might be managed by different groups using different sets of tools.

Assessing Contention

From the perspective of an application running on a VM, the symptoms of contention somewhere in the network normally manifest themselves in the form of high latency and/or dropped packets. The source of the contention might be a switch (either Ethernet or Fibre Channel), a network interface card or the storage array itself. Finding and fixing the problem normally requires the use of a traffic monitor or analyzer tool provided by the switch or SAN vendor, or a third-party management system.

The only situation where the virtualized environment might be the source of the contention is if the host's network interface card has become a bottleneck. This might be the case if the application load on the host is too high or, more likely, if the peak periods in two different applications are occurring simultaneously. Fortunately, such a situation should be a fairly easy to detect and remedy by relocating one of the applications to another host.

Assessing Resource Availability

Assessing how much network bandwidth is available for a new VM cannot be undertaken precisely as some of the traffic that existing VMs generate may flow to systems inside a host and not require network access. Additional hardware based complexity and the inclusion of storage traffic within the network makes this resource area increasingly hard to measure cleanly. Fortunately, network constraints for deploying new VMs are rare and easy to catch if they occur, with any dropped packets serving as an instant identifier of issues.

Factoring in VM Growth

VM growth is considered above in the workflows for some of the critical resources, but it is a sufficiently important consideration to reevaluate here in the aggregate for all resources.

Many, if not most, applications experience growth in usage over time. But others may have achieved a steady state rate of use, while some (the so-called "legacy" applications) may even be in decline. Evening out the distribution of this application mix on each host will serve to avoid problems while minimizing the need to over-allocate resources to accommodate growth. For example, a new application experience a rapid gain in popularity could be paired with several others whose use is in decline.

The planning horizon is also an important consideration here. If planning is a frequent activity, as it can be with purpose-built management tools, the growth factor can be kept relatively small. But if planning occurs on an annual basis, a more generous allocation for growth based on foreseeable and even potential needs may be necessary.

Factoring in Additional Headroom as a Buffer

Headroom is also considered above in the workflows for some of the critical resources, but it is a sufficiently important consideration to reevaluate here in the aggregate for all resources.

It is prudent practice in virtualized environments to leave adequate headroom to avoid problems during peaks in resource demands. VMware recommends, for example, using a headroom of 15-20% on the memory, CPU and storage (for snapshots) in the hosts. Regular monitoring of actual utilization will reveal how well this "rule of thumb" recommendation satisfies particular needs.

As with the allocation for growth, the planning horizon is also a consideration with headroom. Headroom gets consumed as applications grow, causing peaks to become either higher or extended over longer periods—or, more likely, both. If the next capacity planning effort is far in the future, even 20% of headroom may be insufficient.

Measuring a VM's Resource Load Signature

Before counting available VM slots, it is necessary to assess how those slots will be used. And there are a variety of ways to estimate the resource requirements that a new application should require to perform as desired. In the simplest case, the application will already be running on a dedicated server and is now part of a physical-to-virtual (P2V) migration, making it easy to accurately determine the resource requirements. In other cases, the new application may be sufficiently similar to an existing application with a long history of actual utilization, making an accurate estimate possible. These are the easy cases; covered below are the more challenging ones.

Vendor-supplied Application Software

For vendor-supplied applications, *caveat emptor* is in order when determining VM resource requirements. Very few vendors provide specifications for virtualized environments, preferring instead to have their application run on a dedicated server (for obvious and rather self-serving reasons). Virtually every vendor is also notoriously conservative, and to avoid any problems, will recommend a rather generous (over-)allocation of resources. A better way for estimating resource requirements will, therefore, be needed.

Deciding Between Using an Average or Maximum Resource Load

For applications where the optimal resource allocation is unknown and difficult to estimate, the first issue to be considered is whether to employ the average or maximum load. This is where the classic performance vs. cost tradeoff (discussed above) enters the equation. It is also important to make this tradeoff for all five resources: memory, CPU, storage I/O, storage space and the network.

The widely accepted practice here is to use average load for most applications. After all, average utilization rates are one of the reasons virtualization is even possible. The single exception is for mission-critical applications that require high, sustained service levels, for which it is better (at least initially) to use the maximum utilization.

It is important to note that the allocation of headroom discussed above is based on the use of average utilization rates for all or most applications. If resources are allocated based on peak or maximum loads instead, then there is no reason setting aside any headroom.

Testing a VM Being Deployed for the First Time

If all else fails, test the application in a production or development environment. This is the only sure-fire way to know how an application will perform with different VM resource allocations. Granted, testing an application in a meaningful way is easier said than done, and many test set-ups have been known to produce misleading results. But some testing is still better than making a guesstimate with no basis in fact.

Making Adjustments

Yogi Berra famously said: "Making predictions is difficult, especially about the future." So like in war, where the battle plan is inevitably the first casualty, be prepared to discover (perhaps rather quickly) that the estimate of resource requirements is wrong (perhaps significantly). This is not a catastrophe. A VM that is improperly configured for its application can be right-sized, permitting a rapid resolution to the problem. In this respect, new applications are not much different from existing applications that may change in some respect which alters the way it uses different resources. So make a reasonably good estimate of the resource requirements for each new application, and proceed to the final step in the process: calculating the number of VM slots available.

Calculating Available VM Slots

At this point in the process, the current utilization of all five critical resources should be known for all hosts in the affected infrastructure area, and estimates of the resource requirements for all new applications should be available. The diagram below depicts the final step: the workflow for determining VM slot availability.

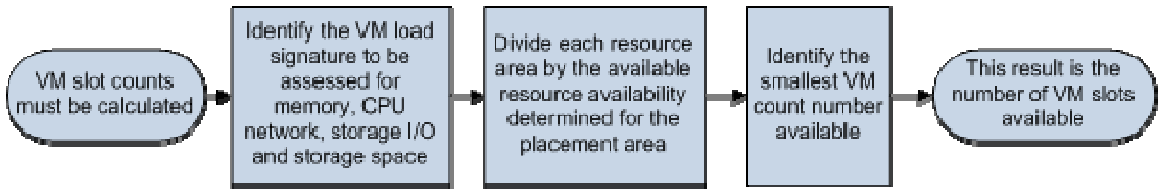

VM slot counts must be calculated
Identify the VM load signature to be assessed for memory, CPU network, storage I/O and storage space
Divide each resource area by the available resource availability determined for the placement area
Identify the smallest VM count number available
This result is the number of VM slots available

Figure 8 –Calculating VM Slot Counts

Find the Amount of Available CPU, Memory, Storage I/O and Storage Capacity

The load signatures for the new applications must now be evaluated against the available resources. Resource availability as described in previous sections, is determined by subtracting actual VM utilization and allocations for forecasted growth and headroom allocation from the host or datastore's total capacity. The load signatures will place a greater or lesser demand on each of the five critical resources, and resource availability will vary by host. At this stage it will be necessary, therefore, to begin considering which hosts will be targeted for which applications.

Divide Expected Resource Usage for the New VM(s) from Available Capacity

The next step is to tale the resource load for the new VM for CPU, memory, storage space, network, and storage I/O, and divide it by the amount of available capacity for each resource area. Results should be rounded down. These calculations will reveal how many VM slots are available for that kind of VM for each kind of resource the VM will need. For example, with 24 GB of VM memory remaining, a 4 GB application will yield six VM slots for that kind of VM in the memory resource. Importantly though, because an application must use all resources, an additional step must be taken to find how many VM slots actually exist.

Identify the Constraining Resource to Determine the VM Slot Count

After all VM slot numbers with available free capacity across memory, CPU, storage I/O, storage and network have been identified, find the lowest slot count out of all the resources. This is the actual amount of VM slots that remain in the infrastructure for that kind of VM.

Whichever resource identified also reveals the constraining resource for further growth and should be further investigated, as buying more of that particular hardware could yield slots for more VMs without having to purchase significant amounts of other hardware.

Additional High Availability Cluster Considerations

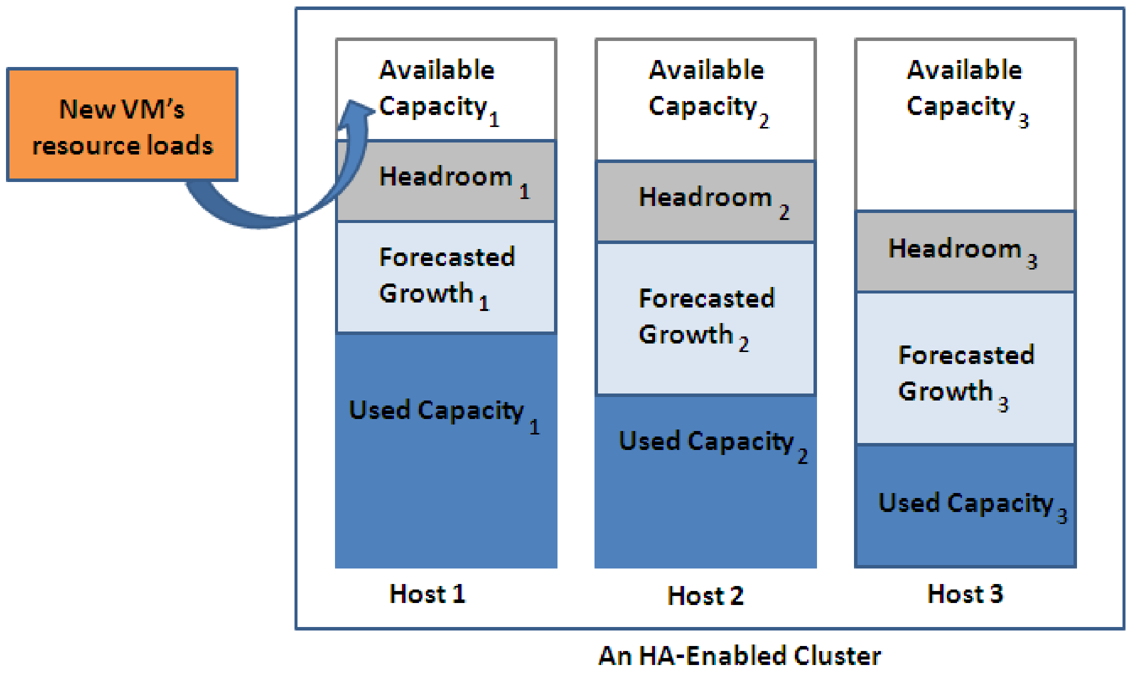
Determining the VM slots available in an HA cluster is somewhat more complicated. The reason is: There needs to be sufficient capacity of all resources within the cluster to handle failovers, which requires identifying the constraining resource for multiple hosts simultaneously.

In an HA cluster environment, settings for shares, reservations and limits must be considered in the assessment. In its set of best practices for clusters, VMware recommends using shares as the primary means to balance loads with the Distributed Resource Scheduler. The company further cautions not to set reservations “too high” because, as noted above, a non-zero reservation dedicates use of that particular resource exclusively to the VM, making it unavailable to all other VMs, including any new ones.

Identify the Constraining Resource to Calculate the VM Slot Count

When following best practices for HA clusters, DRS is able to balance the aggregate load fairly well, even during a failure. Nevertheless, there is always a constraining resource within the cluster that limits its ability to support additional applications. Normally that resource is either available memory or CPU capacity. Identifying the constraining resource gets complicated, however, owing to the number of permutations and combinations of different failover scenarios.

The diagram below shows the situation graphically, taking into account forecasted growth and headroom. The task here is essentially to determine which resource can handle the *lowest* number of new VM slots. Note how the equations address different failover scenarios at the host level. This analysis assumes the use of shares. In this case, Host_1 determines the VM slot availability. The use of reservations would require a similar but more granular analysis of specific resources.



New VM's Resource Load + (Host 1 – Available Capacity$_1$) Must be Less Than (Available Capacity$_2$ + Available Capacity$_3$)

New VM's Resource Load + (Host 2 – Available Capacity$_2$) Must be Less Than (Available Capacity$_1$ + Available Capacity$_3$)

New VM's Resource Load + (Host 3 – Available Capacity$_3$) Must be Less Than (Available Capacity$_1$ + Available Capacity$_2$)

Figure 7 –Determining if Sufficient Resources are Available within HA Clusters

Conclusion

The metrics-based approach to assessing VM slot availability outlined here employs readily available information to make this important task more of a precise science. The results are inevitably more accurate and less disruptive than the alternative, more artistic approach: an iterative, trial-and-error process. The steps outlined in the workflows can be performed manually or by using tools purpose-built for capacity planning. Either way, the process is the same; the difference is the amount of manual effort required.

About VKernel

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.