Performance Troubleshooting for VMware vSphere 4

Introduction

Performance problems can arise in any computing environment. Complex application behaviors, changing demands, and shared infrastructure can lead to problems arising in previously stable environments. Troubleshooting performance problems requires an understanding of the interactions between the software and hardware components of a computing environment. Moving to a virtualized computing environment adds new software layers and new types of interactions that must be considered when troubleshooting performance problems.

Proper performance troubleshooting requires starting with a broad view of the computing environment and systematically narrowing the scope of the investigation as possible sources of problems are eliminated. Troubleshooting efforts that start with a narrowly conceived idea of the source of a problem often get bogged down in detailed analysis of one component, when the actual source of problem is elsewhere in the infrastructure. In order to quickly isolate the source of performance problems, it is necessary to adhere to a logical troubleshooting methodology that avoids preconceptions about the source of the problems.

This document is the first installment in a guide covering performance troubleshooting in a vSphere environment. It uses a guided approach to lead the reader through the observable manifestations of complex hardware/software interactions in order to identify specific performance problems. For each problem covered, it includes a discussion of the possible root-causes and solutions. This first installment covers performance troubleshooting on a single VMware ESX 4.0 host. It focuses on the most common performance problems which affect an ESX host. Future updates will add more detailed performance information, including troubleshooting information for more advanced problems and multi-host vSphere deployments.

Most of the information in this guide can be applied to earlier versions of VMware Virtual Infrastructure and VMware ESX. However, some of the performance metrics accessed using the vSphere Client have been renamed from previous versions, while others were previously only accessible using esxtop. In most cases, the equivalent metrics should be evident upon examination of the appropriate documentation.

The troubleshooting process starts with the top-level troubleshooting flow in Top-Level Troubleshooting Flow. However, the introductory material includes background information that is important for the successful completion of a performance troubleshooting effort. Identifying Performance Problems discusses what we mean by a performance problem, and how to tell when observed behavior is and is not a problem. Performance Troubleshooting Methodology gives an overview of the troubleshooting methodology used in this document. Using This Guide discusses how to use this guide. The guided approach makes using this document different than reading a typical manual. Performance Tools Overview gives an overview of the performance monitoring tools used in the troubleshooting process. Starting with Top-Level Troubleshooting Flow, the remainder of the document covers the process of using the performance monitoring tools to identify observable performance problems, find the root-cause of those problems, and then fix the causes.

This is a living document. Reader comments, questions, and suggestions are encouraged. See Change Information for change and version information.

Contents

Identifying Performance Problems

Performance troubleshooting is a task that is undertaken with the goal of solving performance problems. Therefore, before we begin to discuss performance troubleshooting, we need to clearly define what is, and is not, a performance problem.

The proper way to define performance problems is in the context of an ongoing performance management and capacity planning process. Performance management refers to the process of establishing performance requirements for applications, in the form of Service Level Agreements (SLAs), and then tracking and analyzing the achieved performance to ensure that those requirements are met. A complete performance management methodology would include collecting and maintaining baseline performance data for applications, systems, and subsystems (e.g. storage and network). Capacity planning refers to the process of using modeling and analysis techniques to project the impact of anticipated workload or infrastructure changes on the ability of an infrastructure to satisfy SLAs.

In the context of performance management, a performance problem exists when an application fails to meet its predetermined SLA. Depending on the specific SLA, the failure may be in the form of excessively long response-times, or throughput below some defined threshold. Performance troubleshooting should be undertaken to find the cause and bring performance back within limits defined by the SLAs.

When SLAs or other performance criteria have not been defined, the definition of performance problems becomes more subjective. Baseline performance data, from periods in which performance was deemed acceptable, can be used as a means of quantifying deviations in performance and determining whether problems exist. Ideally, this baseline data should cover the application, the load applied to the application, and the performance characteristics of the server, storage, and network infrastructure. If it is decided that the deviations in application performance are severe enough, then performance troubleshooting should be undertaken to bring performance back within a predetermined range of the baseline.

In environments where no SLAs have been established, and where no baseline data is available, user complaints about slow response-time or poor throughput may lead to the declaration that a performance problem exists. In this situation, there is no objective way of determining whether current performance is problematic. In order to avoid unnecessary investigations, a clear statement of the perceived problem and acceptable solution should be formulated before undertaking performance troubleshooting.

Regardless of which situation exists, it is important to eliminate changes in load as the source of performance problems before investigating the software/hardware infrastructure. Changes in load, such as growth in the number of users, or increased demand from existing users, can cause performance problems in applications that previously had acceptable performance. If a performance management and capacity planning methodology is being followed, baseline data can be used to determine whether changes in application load are responsible for performance degradations. If such data is not available, other avenues of investigation, including customer interviews, should be used to determine whether there have been recent changes in load.

Using a virtualized environment adds one new factor to the definition of performance problems. Often baseline data collected when an application was running on non-virtualized systems is compared to data collected after the application is virtualized. Many factors can lead to differences in performance, including different server hardware, different CPU and memory allocations, sharing of resources with other high-load applications, and different storage subsystems. The following points should be considered when comparing performance between virtualized and non-virtualized environments:

- Performance comparisons should be based on actual application performance metrics, such as response-time or throughput. Comparisons in terms of lower-level performance metrics, such as CPU utilization, are rarely valid when moving from a physical to virtual environment.

- Performance comparisons between virtual and non-virtual environments are only valid when the same underlying infrastructure is used. This includes the same type and number of CPU cores, the same amount of memory, and storage which is either the same or has comparable performance characteristics.

See the VMware technical document *Performance Best Practices and Benchmarking Guidelines* for additional information on benchmarking performance in a virtualized environment.

Performance Troubleshooting Methodology

Once it has been determined that a performance problem exists, it is necessary to follow a logical methodology for finding and fixing the cause of the problem. In this section we describe the troubleshooting methodology used in this guide for finding and fixing performance problems in a vSphere environment

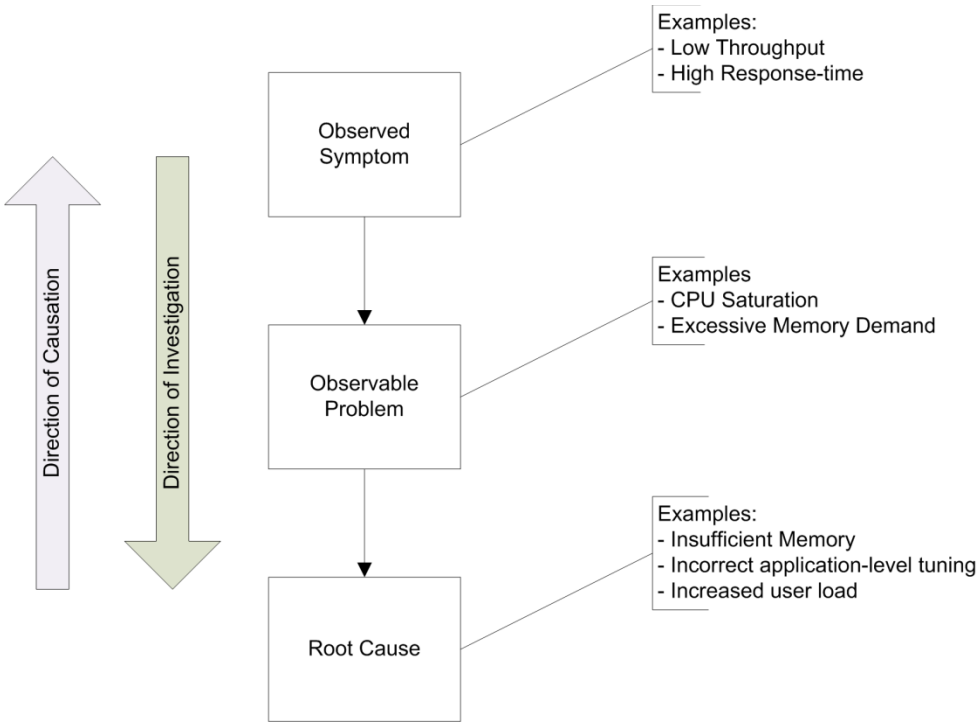
In discussing our performance troubleshooting methodology, we use the following terms:

- **Observed Symptoms:** These are the observed effects that lead to the decision that a performance problem exists. They are based on high-level metrics such as throughput or response-time. Ideally the presence/absence of a problem is defined by an SLA or other set of performance targets or baselines.

- **Observable Problem:** These are the problems that can be identified in the lower-level infrastructure by the values of specific performance metrics. The performance metrics are typically provided by performance monitoring tools. An observable problem may be directly causing the symptoms, but there is typically something more fundamental that is causing the problem to occur.

- **Root Cause:** The root cause is the ultimate source of the observable problems. It may be a configuration issue, a contention for resources, application tuning, etc. Root causes often cannot be directly observed by monitoring tools, but instead must be inferred by the presence of observable problems. Finally identifying a root-cause may require an iterative tuning effort.

Figure 1. Relationship among Symptoms, Problems, and Cause



A performance troubleshooting methodology must provide guidance on how to find the root-cause of the observed performance symptoms, and how to fix the cause once it is found. To do this, it must answer the following questions:

1. How do we know when we are done?
2. Where do we start looking for problems?
3. How do we know what to look for to identify a problem?
4. How do we find the root-cause of a problem we have identified?
5. What do we change to fix the root-cause?
6. Where do we look next if no problem is found?

The first step of any performance troubleshooting methodology must be deciding on the criteria for successful completion. If SLAs or other baselines exist, they can be used to generate the success criteria. Otherwise, application-specific knowledge, an understanding of user expectations, and an understanding of available resources can be used to help determine the criteria. The process of setting SLAs or selecting appropriate performance goals is beyond the scope of this document, but it is critical to the success of the troubleshooting process that some stopping criteria be set. In the absence of defined goals, performance troubleshooting can turn into an endless performance-tuning process.

Having decided on an end-goal, the next step in the process is deciding where to start looking for observable problems. There can be many different places in the infrastructure to start an investigation, and many different ways to proceed. The goal of our performance troubleshooting methodology is to select at each step the component which is most likely to be responsible for the observed problems. In our experience, a large percentage of performance problems in a vSphere environment are cause by a small number of common causes. As a result, the method used in this document is to follow a fixed-order flow through the most common observable performance problems. For each problem, we specify checks on specific performance metrics made available by the vSphere performance monitoring tools. These checks are embodied in flow-charts that describe each step necessary to access the metrics, and values that indicate the presence or absence of the problem. If a problem is identified, the problem checks direct the reader to the section of this document that discusses the possible root-causes for the problem, and possible solutions.

Once a problem has been identified and resolved, the performance of the environment should be re-evaluated in the context of the completion criteria defined at the start of the process. If the criteria are satisfied, then the troubleshooting process is complete. Otherwise the problem checks should be followed again to identify additional problems. As in all performance tuning and troubleshooting processes, it is important to fix only one problem at a time, so that the impact of each change can be properly understood.

Using This Guide

Unlike a typical book or manual, this guide is not intended to be read linearly from start to finish. Instead, it is based around the hierarchical troubleshooting flow-charts which begin in Top-Level Troubleshooting Flow. Following the steps indicated by the flow-charts will lead the reader through checks for most common sources of performance problems in a vSphere environment. Checks that indicate the presence of a problem then point to relevant discussions of possible causes and solutions.

The flow-charts are arranged hierarchically, with each level having a specific intent.

- The top-level flow-chart leads through identification of candidate areas for investigation based on information about the problem and the computing environment. The nodes in this flow point off to one or more mid-level flow-charts for each general area. At present, this flow is limited to a single ESX host, but it will be expanded in future updates to this document.

- The mid-level flow-charts lead through the problem checks for specific observable problems in a given area. All of the problem checks covered by a mid-level flow use a common set of monitoring tools. Each node in a mid-level flow-chart points off to a bottom-level flow-chart which contains the actual problem check.

- The bottom-level flow-charts are the problem checks for specific observable problems. The nodes in these flow-charts will direct you to perform specific actions or observe the values of specific performance metrics. If the outcome of the check is to confirm that the problem exists or may exist in the environment, the flow points to the section of this document that discusses possible causes and solutions.

The number of flow-charts and steps in this troubleshooting process may seem intimidating at first. However, most of the steps require checking only a small number of easily accessed performance metrics, and can be performed in a few minutes using the facilities provided by the vSphere client. With a little practice, performing these checks requires very little time, and can save a lot of effort when trying to solve performance problems in a vSphere environment.

Performance Tools Overview

Checking for observable performance problems requires looking at values of specific performance metrics, such as CPU utilization or disk response-time. vSphere provides two main tools for observing and collecting performance data. The vSphere Client, when connected directly to an ESX host, can display real-time performance data about the host and VMs. When connected to a vCenter server, the vSphere Client can also display historical performance data about all of the hosts and VMs managed by that server. For more detailed performance monitoring, esxtop and resxtop can be used, from within the Service Console or remotely, respectively, to observe and collect additional performance statistics.

The vSphere Client will be our primary tool for observing performance and configuration data for ESX hosts. Complete documentation on using obtaining and using the vSphere client can be found in the *vSphere Basic System Administration* guide. The advantages of the vSphere Client are that it is easy to use, provides access to the most important configuration and performance information, and does not require high levels of privilege to access the performance data.

The esxtop and resxtop utilities provide access to detailed performance data from a single ESX host. In addition to the performance metrics available through the vSphere Client, they provide access to advanced performance metrics that are not available elsewhere. The advantages of esxtop/resxtop are the amount of available data and the speed with which they make it possible to observe a large number of performance metrics. The disadvantage of esxtop/resxtop is that they require root-level access privileges to the ESX host. Documentation covering accessing and using esxtop and resxtop can be found in the *vSphere Resource Management Guide*.

Performance troubleshooting in a vSphere environment should use the performance monitoring tools provided by vSphere rather than tools provided by the guest OS. The vSphere tools are more accurate than tools running inside a guest OS. While guest OS performance-monitoring tools are more accurate in vSphere than in previous releases of VMware ESX, there are still situations, such as when CPU resources are over-commited, that can lead to inaccuracies in in-guest reporting. In addition, the accuracy of in-guest tools also depends on the guest OS and kernel version being used. As a result, it is best to use the vSphere provided tools when actively investigating performance issues.

Top-Level Troubleshooting Flow

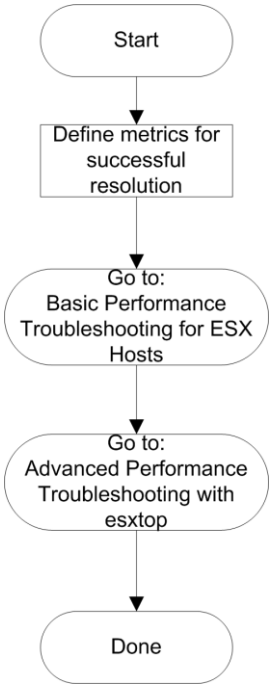
The performance troubleshooting process covered by this guide starts with the top-level flow-chart presented in this section. Most of the nodes in this top-level flow point to a more detailed mid-level flow-chart.

The top-level troubleshooting flow is shown in Figure 2. The first step in the process is to define criteria for successful resolution of the performance problems. See the discussions in Identifying Performance Problems and Performance Troubleshooting Methodology for additional information on this topic.

Once the success criteria have been defined, the next step is to follow the Basic Troubleshooting flow contained in Basic Performance Troubleshooting for VMware ESX. This flow leads through problem checks of the most common observable performance-problems using information accessible through the vSphere client. This is followed by the Advanced Troubleshooting flow in Advanced Performance Troubleshooting with esxtop. This flow looks for less common problems using performance data only accessible through esxtop.

The troubleshooting flows contained in this document do not cover all possible performance problems in a vSphere environment. If you are unable to resolve the problem using these flows, more in-depth performance information should be consulted.

Figure 2. Top-Level Troubleshooting Flow



Basic Performance Troubleshooting for VMware ESX

Overview

This section covers the basic steps for investigating a performance problem on a single ESX host or VM. It walks the reader through checks for common performance problems using information that is accessible to most vSphere administrators.

The basic troubleshooting flow is presented in Basic Troubleshooting Flow. This flow gives a suggested order for checking for common performance-related problems. The checks for each of those problems are given in Basic Problem Checks.

Basic Troubleshooting Flow

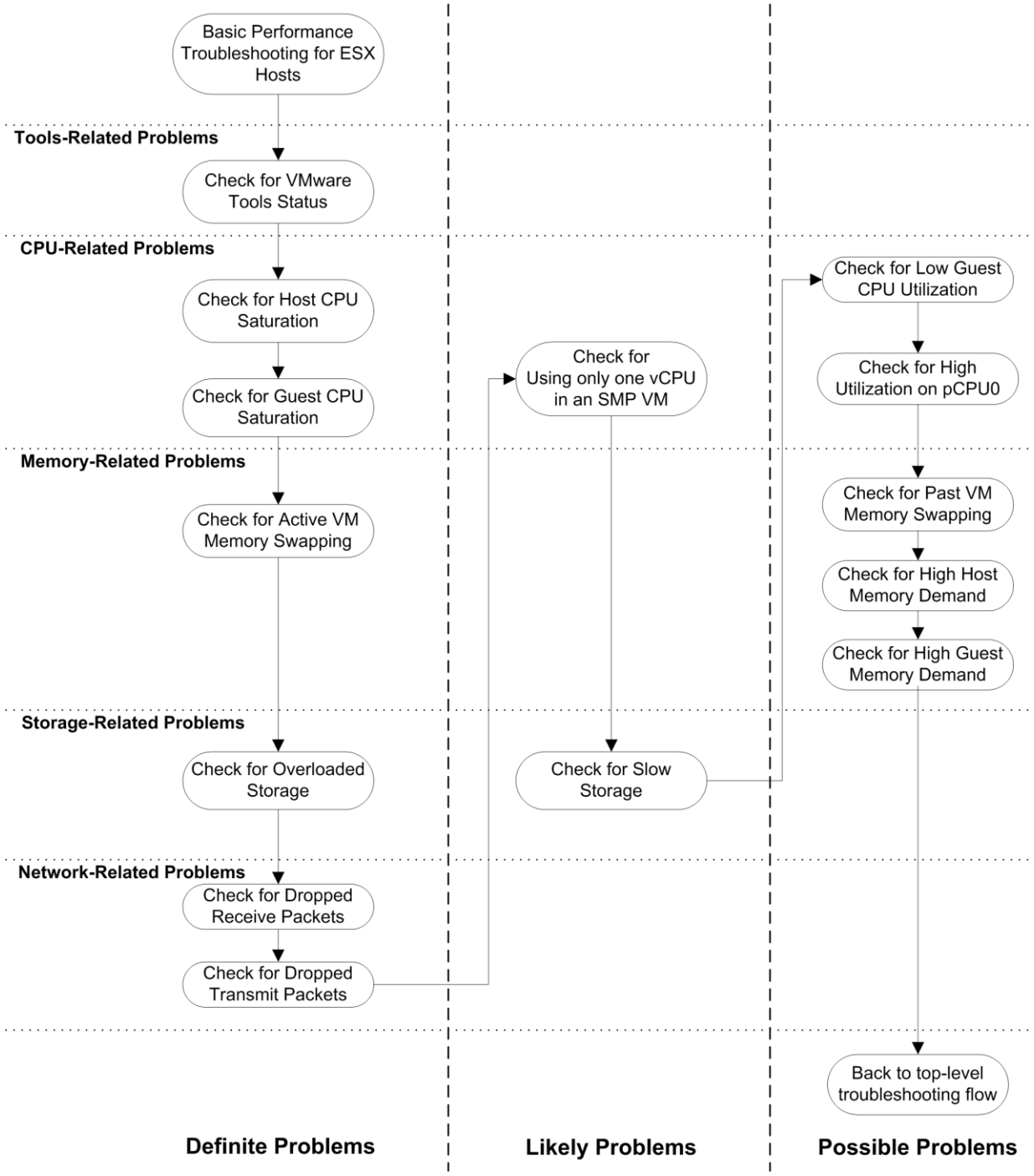
The basic troubleshooting flow is shown in Figure 3. This flow does not include problem checks for all possible performance problems. The problem checks in this flow cover the most common or serious problems which cause performance issues on ESX hosts. Additional problem checks will be added to this flow as this document is updated.

In the basic troubleshooting flow we distinguish among three categories of observable problems: definite, likely, and potential problems. Definite problems are conditions that, if they exist, will have a direct impact on observed performance, and should be corrected. Likely problems are conditions that in most cases will lead to performance problems, but that in some circumstances may not require correction. Potential problems are those conditions which may be indicators of the causes of performance problems, but may also reflect normal operating conditions. Likely and potential problems require additional investigation to determine whether they are causing the observed performance symptoms.

Within each category, the problem checks are organized by general system area. This simplifies checking for problems which might use similar performance metrics.

Flow-charts giving the problem checks for each observable problem covered by this flow are contained in Basic Problem Checks.

Figure 3. Basic Troubleshooting Flow for a VMware ESX Host



Basic Problem Checks

This section contains the problem checks for each of the observable problems covered by the Basic Troubleshooting flow. Performing these problem checks requires looking at the values of specific performance metrics using the performance monitoring capabilities of the vSphere Client. Note that the threshold values specified in these checks are only guidelines based on past experience. The best values for these thresholds may vary depending on the sensitivity of a particular configuration or set of applications to variations in system load. If the values are close to, but

not at, the specified values, it may be worthwhile to read the associated cause and solution discussion to better understand the issues involved.

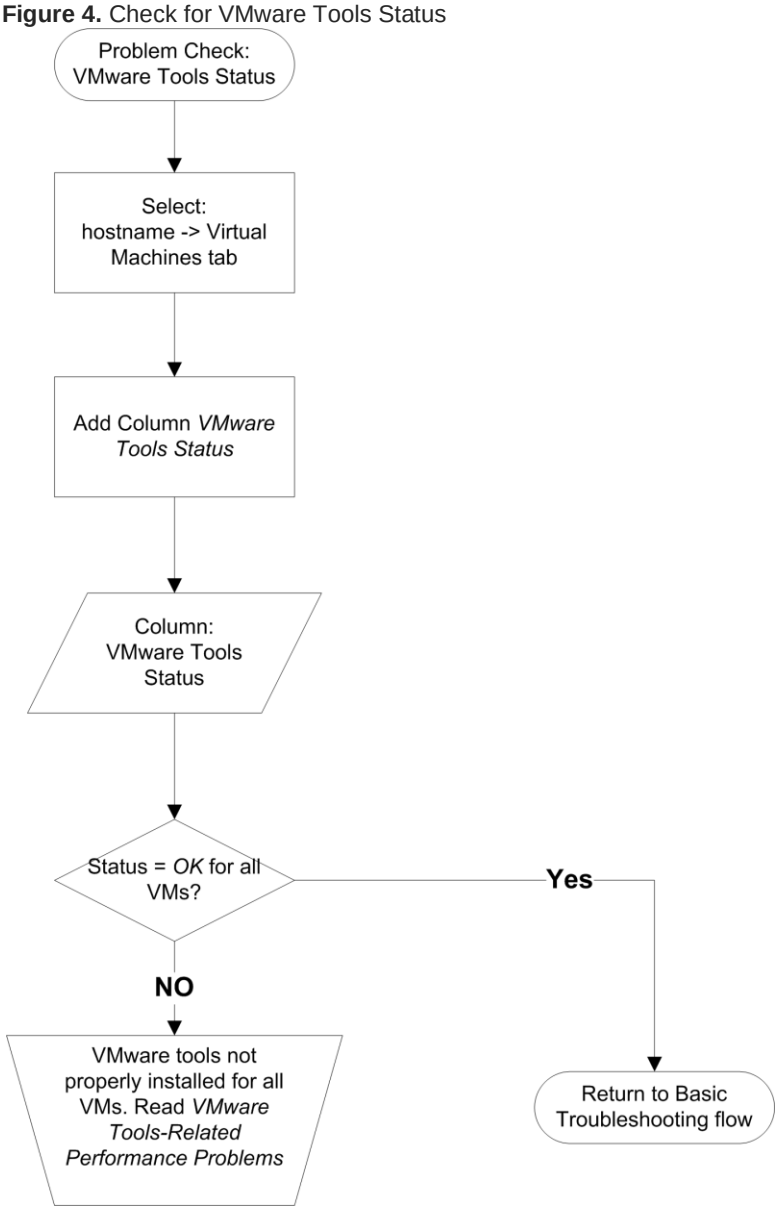
All of the problem checks use data available through the vSphere Client connected either to an ESX host or to a vCenter server. They assume the use of real-time statistics while problem is occurring. Some of these checks may also be possible using historical performance data available when connected to vCenter.

If using the vSphere Client connected directly to an ESX host, ignore the steps that say to switch to the Advanced performance charts. These are the default charts when connected directly to an ESX host.

Check for VMware Tools Status

1. Check tools status
    a. Select the host, then the Virtual Machines tab.
    b. Right-click on the column header, and select VMware Tools Status
    c. Is the status is OK for all powered-on VMs?
        - Yes: Return to Basic Troubleshooting flow.
        - No: If the status is Not Running or Out of date for any VM, go to VMware Tools-Related Performance Problems, for a discussion of possible causes and solutions.

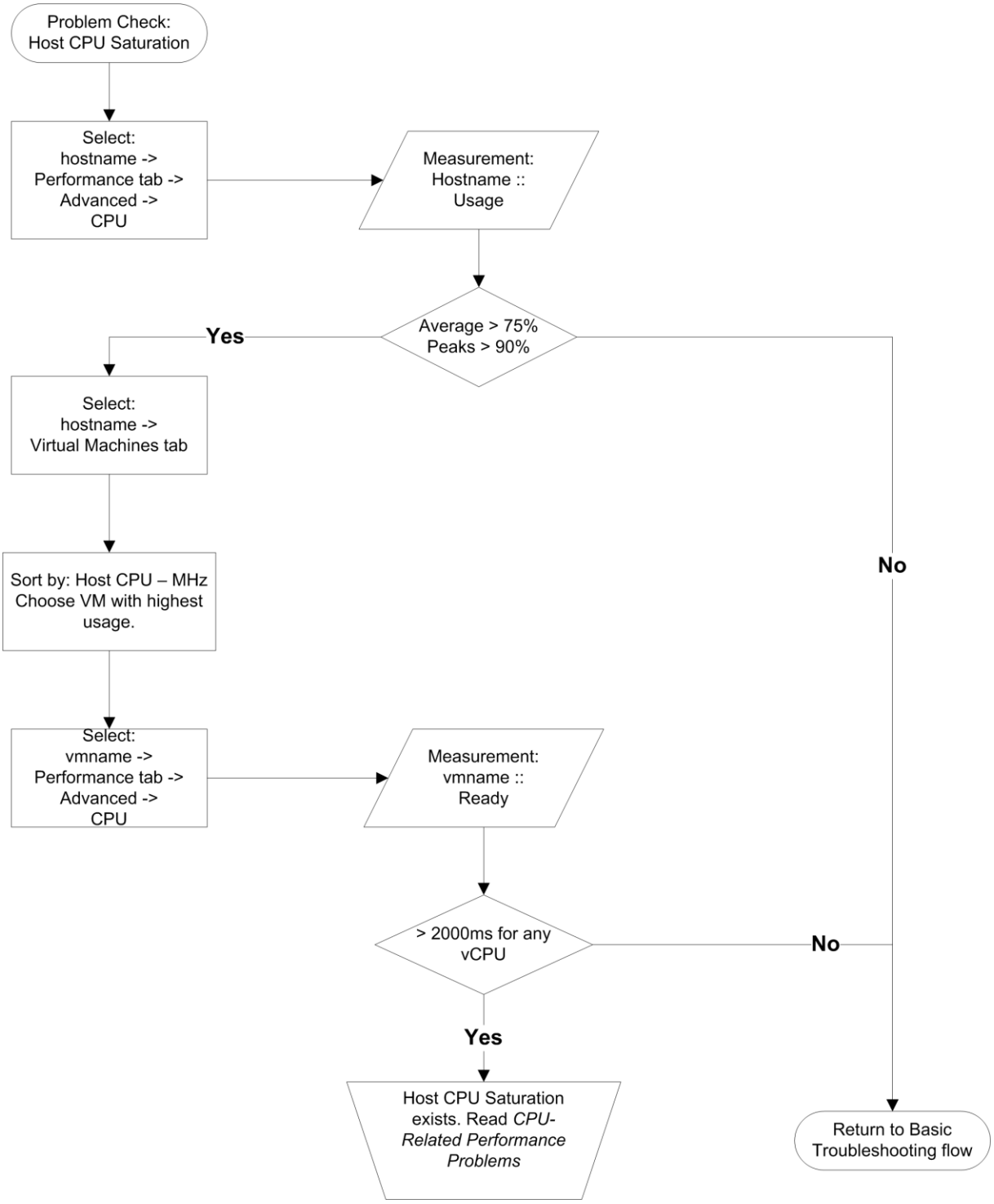
Figure 4. Check for VMware Tools Status



Check for Host CPU Saturation

1. Check for high CPU usage
    a. Select the host, then the Performance tab -> Advanced, then Switch to: CPU
    b. Look at measurement Usage for the hostname object.
    c. Is the average above 75%, or are there peaks in the graph above 90%?
        - Yes: Possible Host CPU Saturation. Go to step 2b to check for high Ready Time.
        - No: Host CPU Saturation is not present. Return to the Basic Troubleshooting Flow.
2. Check for high Ready Time
    a. If the performance problem is specific to a VM, use that VM in the following steps. If not, Select the host, then the Virtual Machines tab. Click on the Host CPU – MHz header to sort by that column. Note the name of the VM with the highest CPU usage.
    b. Select the VM, then the Performance tab, then Switch to: CPU
    c. Look at measurement Ready for all objects. In this case the objects represent the vCPU numbers for the VM. You may need to Change Chart Options in order to view the Ready measurement.
    d. Is Ready greater than 2000ms for any vCPU object?
        - Yes: Host CPU Saturation exists. Go to CPU-Related Performance Problems, for a discussion of possible causes and solutions.
        - No: Host CPU Saturation is not present. Return to the Basic Troubleshooting Flow.

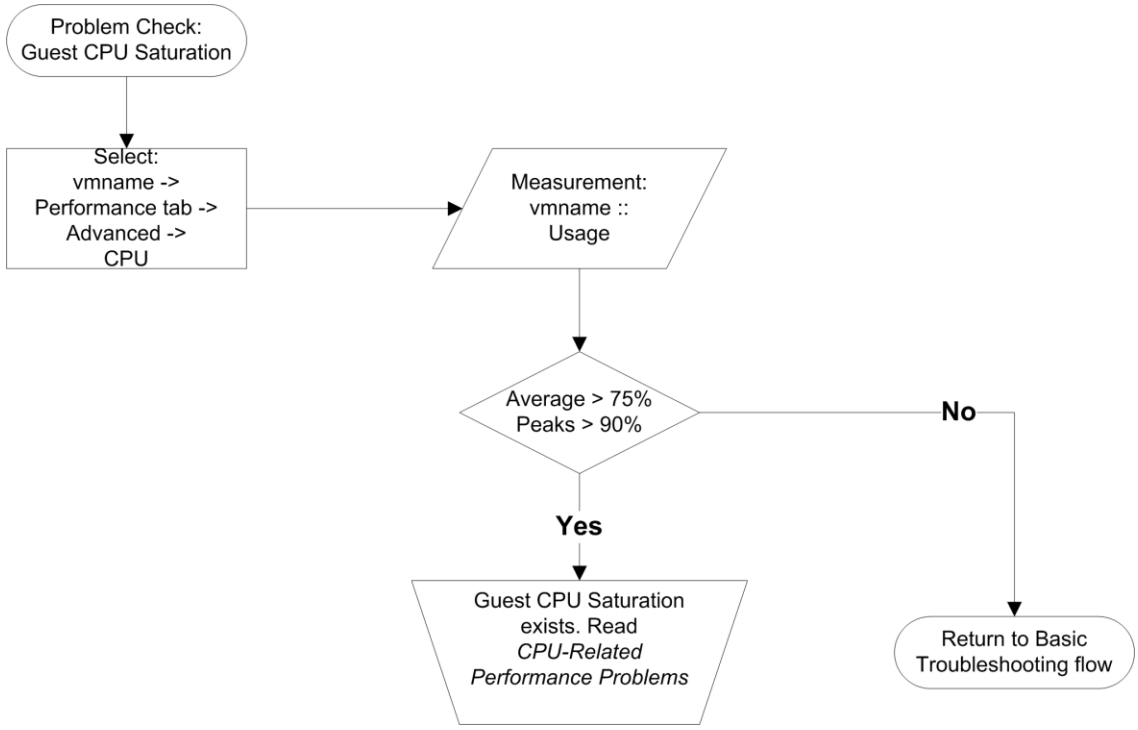
Figure 5. Check for Host CPU Saturation



Check for Guest CPU Saturation

1. Check CPU Usage
    a. If the performance problem is specific to a VM, use that VM in the following steps. If not, Select the host, then the Virtual Machines tab. Click on the Host CPU – MHz header to sort by that column. Note the name of the VM with the highest CPU usage.
    b. Select the VM, then the Performance tab, then Advanced, then Switch to: CPU
    c. Look at measurement Usage for the VM-name object.
    d. Is the average above 75%, or are there peaks in the graph above 90%?
        - Yes: Guest CPU Saturation exists. Go to CPU-Related Performance Problems, for a discussion of possible causes and solutions. If the performance problem is affecting the entire host. Repeat this check for other VMs on the host.
        - No: Guest CPU Saturation is not present. Return to the Basic Troubleshooting flow.

Figure 6. Check for Guest CPU Saturation



Check for Active VM Memory Swapping

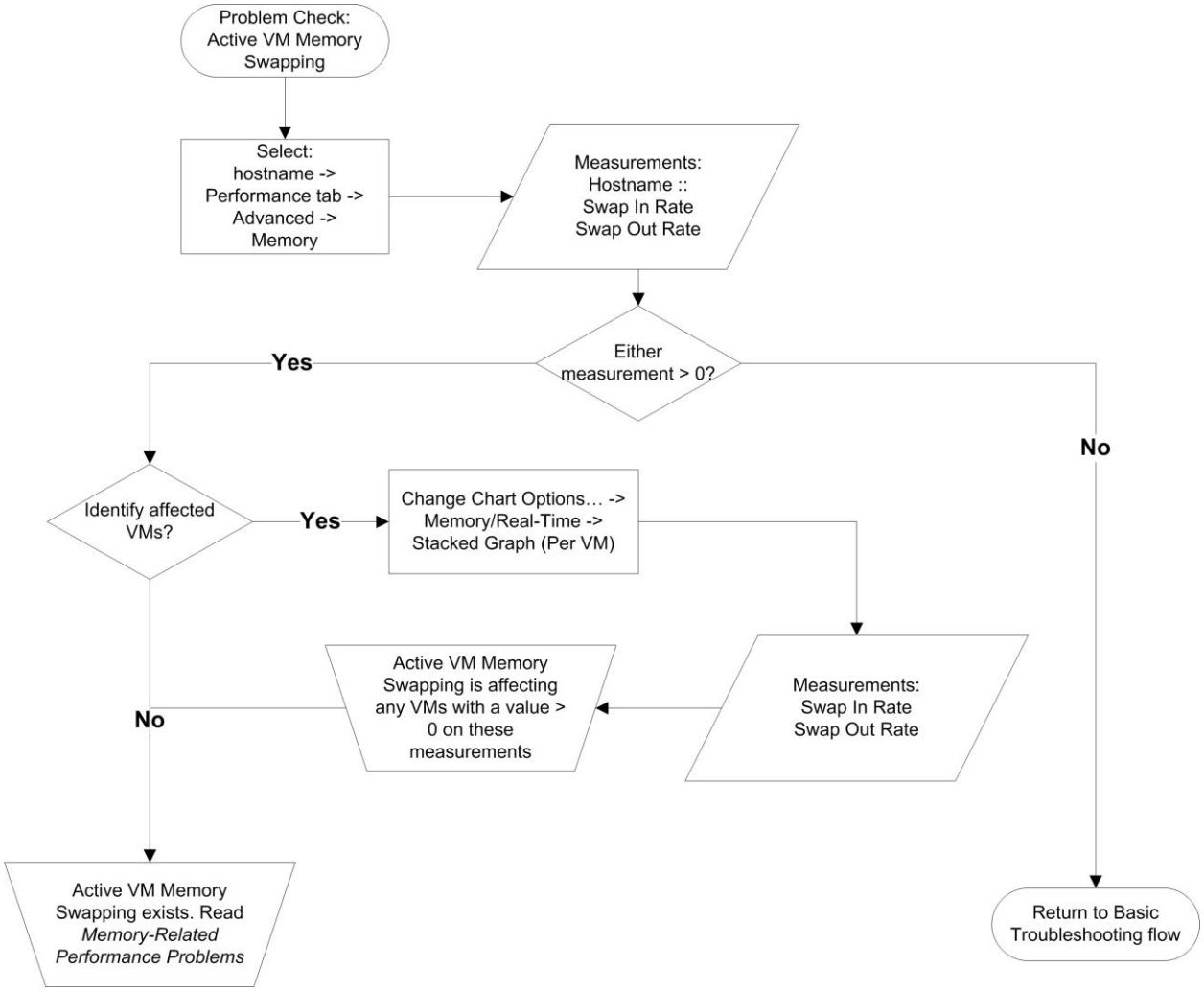
Excessive memory demand can cause severe performance problems for one or more VMs on an ESX host. When ESX is actively swapping the memory of a VM in and out from disk, the performance of that VM will degrade. The overhead of swapping a VM's memory in and out from disk can also degrade the performance of other VMs.

1. Check for active swapping
    a. Select the host, then the Performance tab, then Advanced, then Switch to: Memory
    b. Look at measurements Swap In Rate and Swap Out Rate for the hostname object. You may need to Change Chart Options in order to view these measurements.
    c. Are either of these measurements greater than 0 any time during the displayed period?
        - Yes: The ESX host is actively swapping VM memory. Go to Memory-Related Performance Problems, for a discussion of possible causes and solutions. In order to determine whether this is directly affecting a particular VM, go to step b.

- No: The ESX host is not currently swapping VM memory. Return to the Basic Troubleshooting flow.

2. Check for active swapping in a VM
   a. Select the host, then the Performance tab, then Advanced, then Switch to: Memory
   b. Select Change Chart Options, then select Memory/Real-Time, then change the Chart Type to Stacked Graph (Per VM). Select all VMs.
   c. One at a time, look at the measurements Swap In Rate and Swap Out Rate for all VMs.
   d. VM Memory Swapping is directly affecting those VM's with values greater than 0 on any of these measurements. Go to Memory-Related Performance Problems, for a discussion of possible causes and solutions. Active VM Memory Swapping is not currently directly affecting the other VMs.

Figure 7. Check for Active VM Memory Swapping



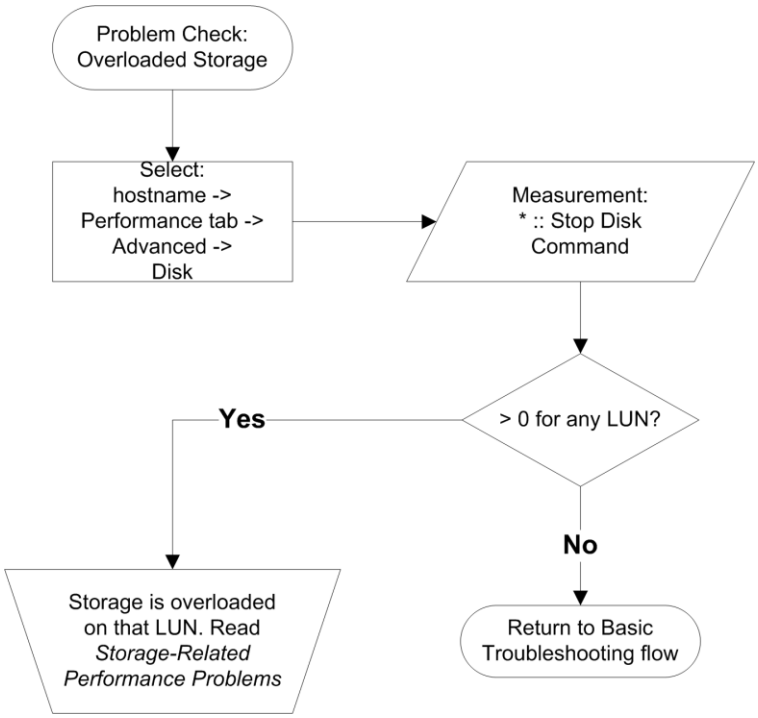
Check for Overloaded Storage

Severely overloaded storage can be the result of a large number of different issues in the underlying storage layout or infrastructure. In turn, overloaded storage can manifest in many different ways depending on the applications running in the VMs.

1. Check for command aborts
   a. Select the host, then the Performance tab, then Advanced, then Switch to: Disk
   b. Look at the measurement Stop Disk Command for all LUN objects.

c. Is the value greater than 0 on any LUN?
   - Yes: Storage is overloaded on that LUN. Go to Storage-Related Performance Problems, for a discussion of possible solutions.
   - No: Storage is not severely overloaded. Return to the Basic Troubleshooting flow.

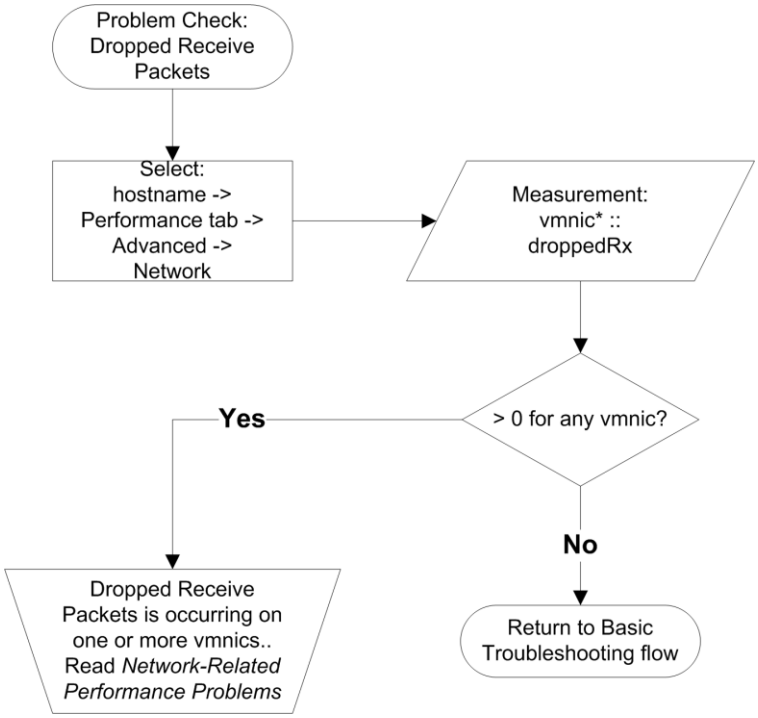
Figure 8. Check for overloaded storage



Check for Dropped Receive Packets

1. Check for dropped receive packets
   a. Select the host, then the Performance tab, then Advanced, then Switch to: Network
   b. Look at the measurement droppedTx for all vmnic objects.
   c. Is the value greater than 0 on any vmnic?
      - Yes: Transmit packets are being dropped on one or more vmnics. Go to Network-Related Performance Problems, for a discussion of possible solutions.
      - No: Transmit packets are not being dropped. Return to the Basic Troubleshooting flow.

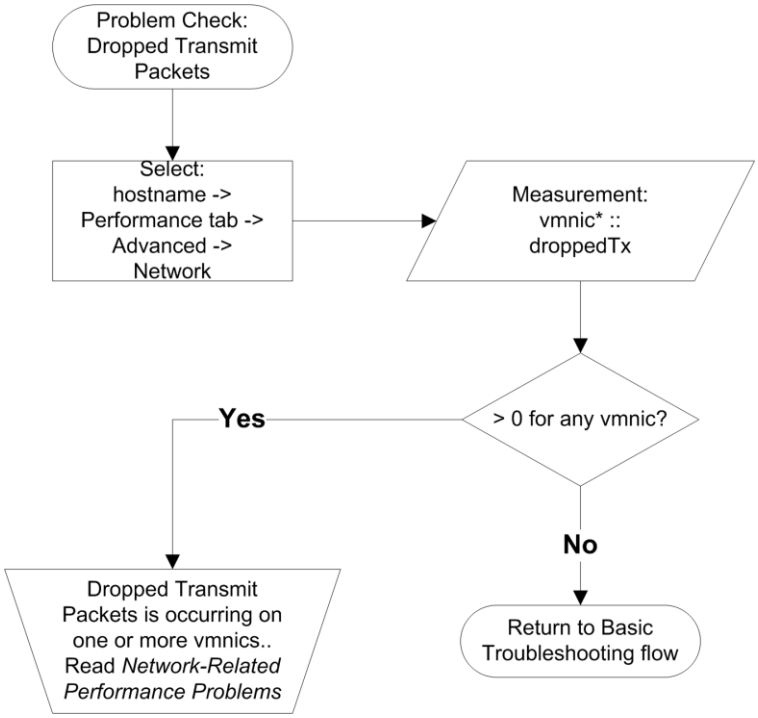
Figure 9. Check for Dropped Receive Packets



Check for Dropped Transmit Packets

1. Check for dropped receive packets
    a. Select the host, then the Performance tab, then Advanced, then Switch to: Network
    b. Look at the measurement droppedTx for all vmnic objects.
    c. Is the value greater than 0 on any vmnic?
        - Yes: Transmit packets are being dropped on one or more vmnics. Go to Network-Related Performance Problems, for a discussion of possible solutions.
        - No: Transmit packets are not being dropped. Return to the Basic Troubleshooting flow.

Figure 10. Check for Dropped Transmit Packets

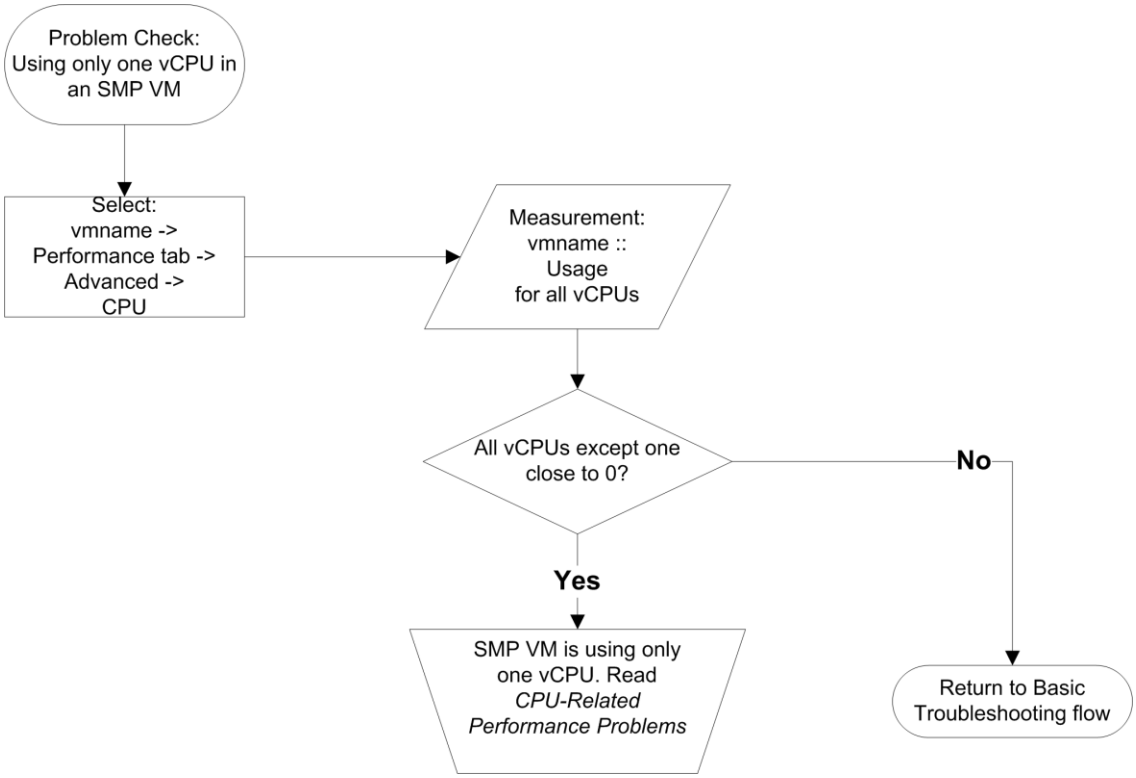


Check for Using only one vCPU in an SMP VM

If a VM configured with more than one vCPU is experiencing performance problems, it may be that the guest OS running in the VM is not properly configured to use all of the vCPUs. This check will help identify whether this problem is occurring.

1. Check vCPU Usage
    a. Select the VM, then the Performance tab, then Advanced, then Switch to: CPU
    b. Look at measurement Usage for the all vCPU objects
    c. Is the usage for all vCPUs except one close to 0?
        - Yes: The SMP VM is using only one vCPU. Go to CPU-Related Performance Problems, for a discussion of possible causes and solutions.
        - No: Return to the Basic Troubleshooting flow.

Figure 11. Check for using only one vCPU in an SMP VM

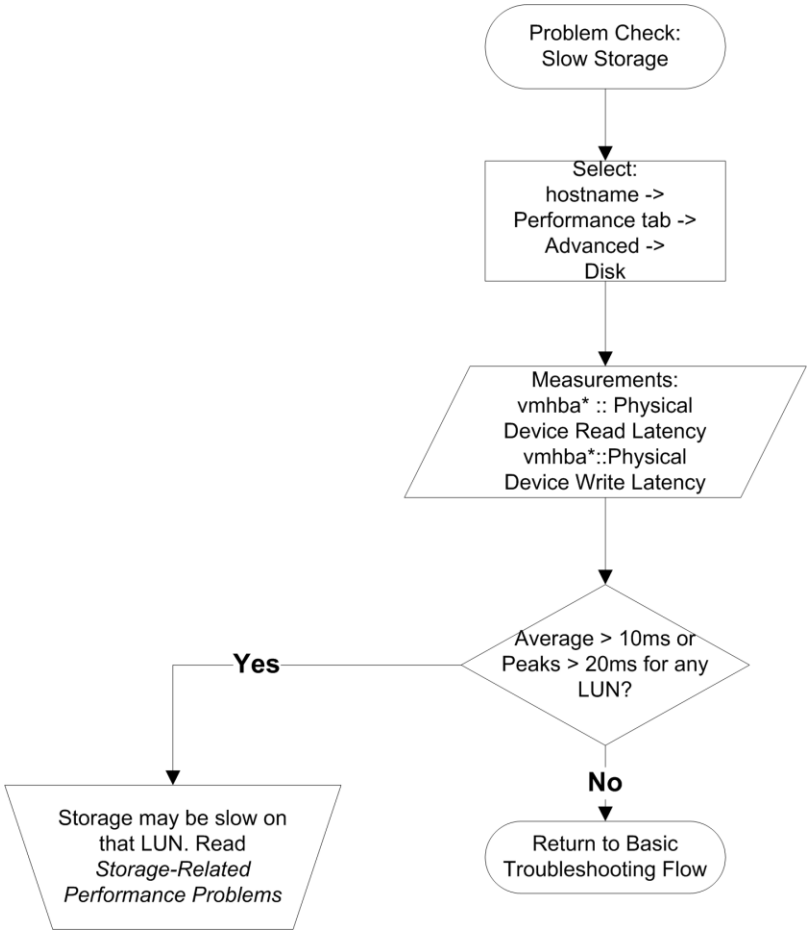


Check for Slow Storage

Slow storage can be the result of a large number of different issues in the underlying storage layout or infrastructure. In turn, slow or overloaded storage can manifest in many different ways depending on the applications running in the VMs.

1. Check for high disk latency
    a. Select the host, then the Performance tab, then Advanced, then Switch to: Disk
    b. Look at measurements Physical Device Read Latency and Physical Device Write Latency for all LUN objects.
    c. For these measurements, is the average above 10ms, or are there peaks above 20ms, for any vmhba?
        - Yes: Storage may be slow or overloaded on that vmhba. Note that these latency values are provided only as points at which further investigation is justified. Expected disk latencies will depend on the nature of the storage workload (e.g. read/write mix, randomness, and I/O size) and the capabilities of the storage subsystems. Read Storage-Related Performance Problems, for further investigation and a discussion of possible causes and solutions.
        - No: Storage does not appear to be slow or overloaded. Return to the Basic Troubleshooting flow.

Figure 12. Check for slow storage

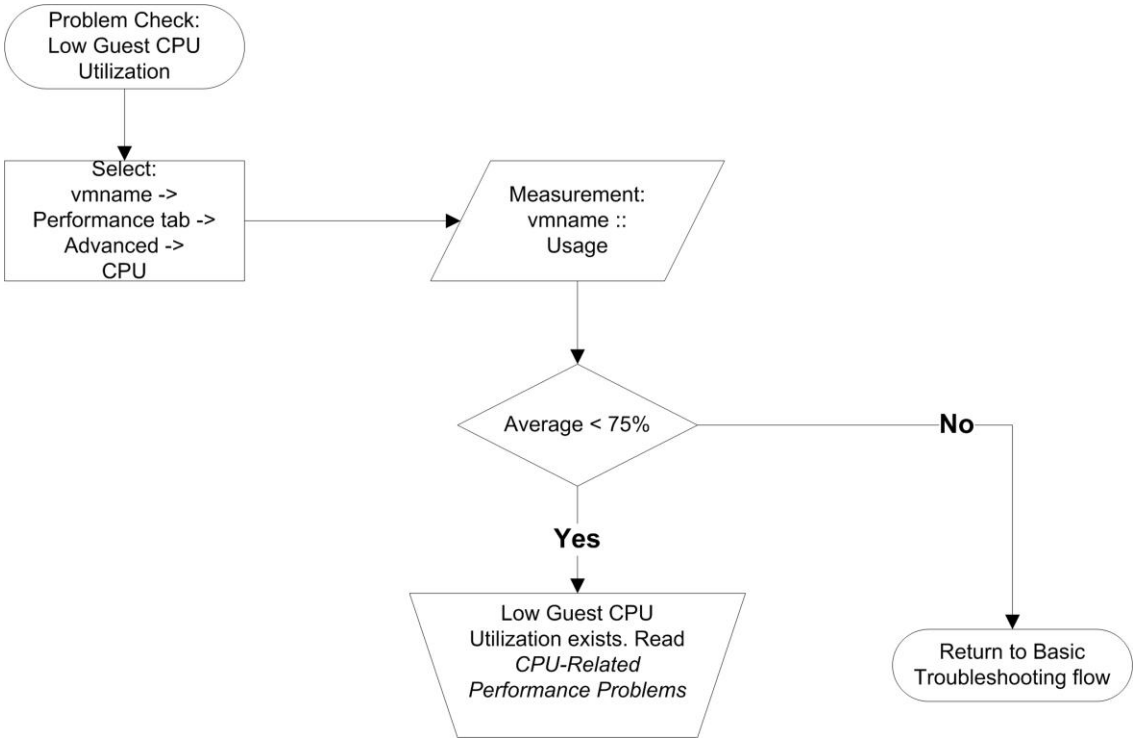


Check for Low Guest CPU Utilization

If a VM is experiencing performance problems even though its CPU utilization is low, there may be some configuration problem or external cause of the poor performance.

1. Check vCPU Usage
    a. Select the VM, then the Performance tab, then Advanced, then Switch to: CPU
    b. Look at measurement Usage for the VM-name object.
    c. Is the average below 75%?
        - Yes: Guest CPU utilization is low. This may indicate one of a number of underlying causes for the performance problems. Go to CPU-Related Performance Problems, for a discussion of possible causes and solutions. If the performance problem is affecting the entire host. Repeat this check for other VMs on the host.
        - No: VM CPU Saturation is not present. Return to the Basic Troubleshooting flow.

Figure 13. Check for Low Guest CPU Utilization

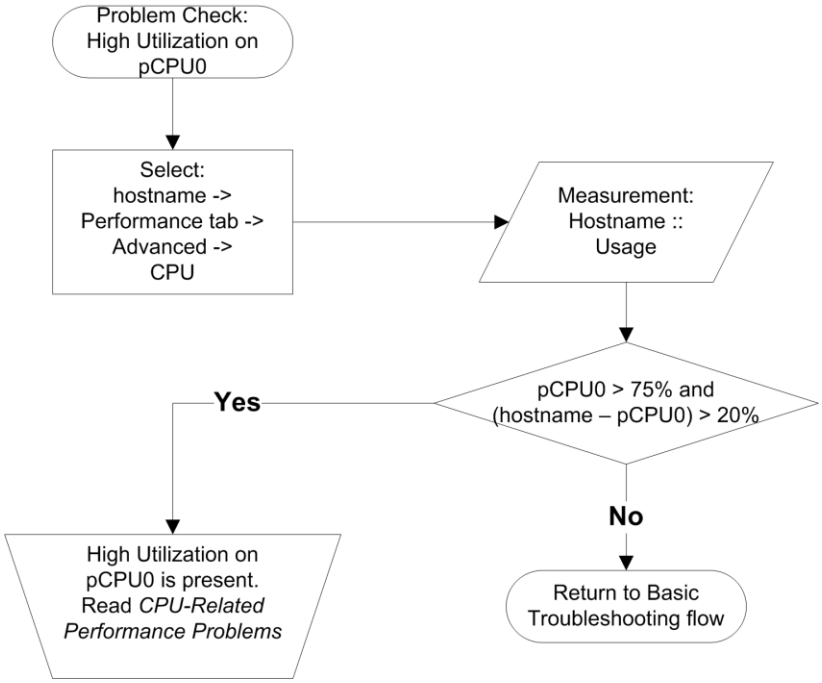


Check for High Utilization on pCPU0

In VMware ESX, excessive activity in the Service Console can in rare cases impact the performance of running VMs. This does not apply to the ESXi edition.

1. Check for high CPU usage on pCPU0
    a. Select the host, then the Performance tab -> Advanced, then Switch to: CPU
    b. Look at measurement Usage for the hostname and CPU 0 objects.
    c. Is the usage on pCPU0 above 75%, and is it more than 20% greater than the overall host usage?
        - Yes: Possible High utilization on pCPU0. Go to CPU-Related Performance Problems, for a discussion of possible causes and solutions.
        - No: High Utilization on pCPU0is not present. Return to the Basic Troubleshooting Flow.

Figure 14. Check for High Utilization on pCPU0

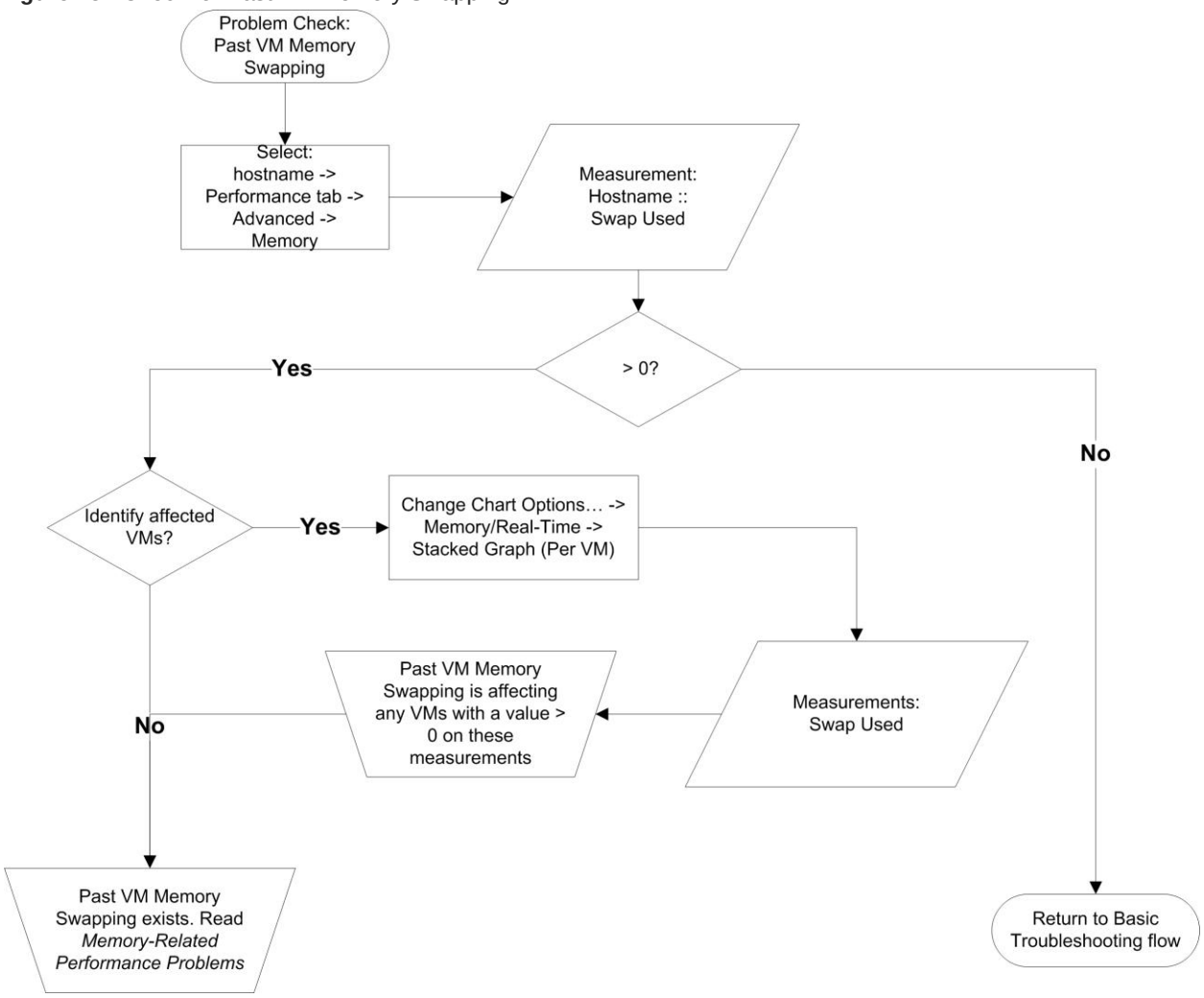


Check for Past VM Memory Swapping

In response to high memory usage, an ESX host may swap VM memory out to swap files on disk. In some instances, that memory will remain swapped out to disk even though the host is no longer actively swapping. This will occur only if no VM is actively using the swapped memory. As a result, past swapping is unlikely to be the cause of ongoing performance problems. However, it may be an indication of the cause of past performance problems.

1. Check for past swapping
    a. Select the host, then the Performance tab, then Advanced, then Switch to: Memory
    b. Look at measurements Swap Used for the hostname object. You may need to Change Chart Options in order to view these measurements.
    c. Is this measurement greater than 0?
        - Yes: The ESX host has swapped VM memory at some point in the past. Go to Memory-Related Performance Problems, for a discussion of possible causes and solutions. In order to determine which VM's memory has been swapped, go to step b.
        - No: The ESX host does not currently have any swapped VM memory. Return to the Basic Troubleshooting flow.
2. Check for swapped memory in a VM
    a. Select the host, then the Performance tab, then Advanced, then Switch to: Memory
    b. Select Change Chart Options, then select Memory/Real-Time, then change the Chart Type to Stacked Graph (Per VM). Select all VMs.
    c. Look at the measurement Swap Used for all VMs.
    d. The ESX host has swapped memory from those VM's with values greater than 0 on this measurement. Go to Memory-Related Performance Problems, for a discussion of possible causes and solutions.

Figure 15. Check for Past VM Memory Swapping



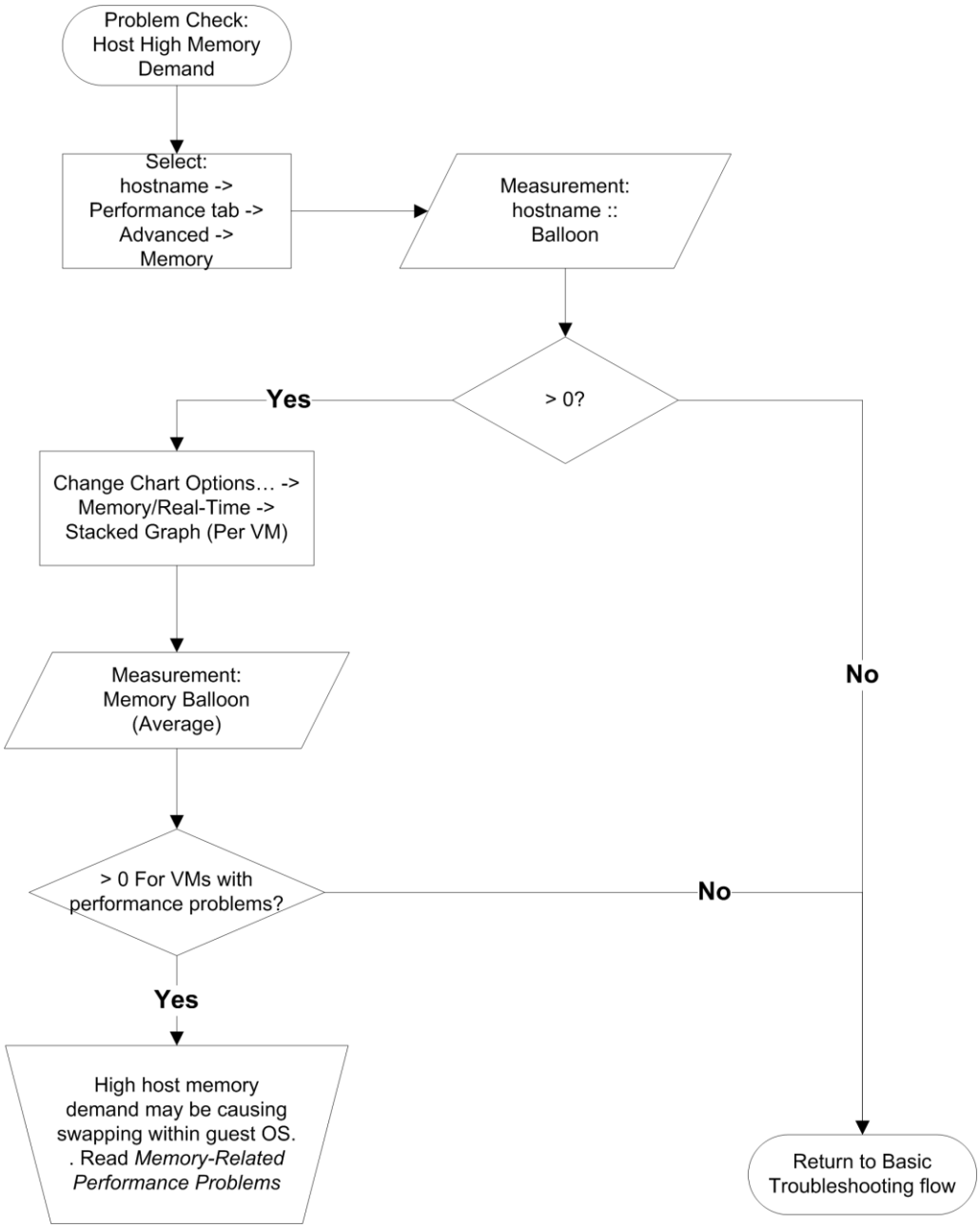
Check for High Host Memory Demand

High memory demand does not always result in performance problems. However, if the ESX reclaims memory from a VM that needs the memory, swapping within the guest OS can result in performance degradation. This condition usually manifests itself as performance problems on individual VMs on the host.

1. Check for ballooning on the host
    a. Select the host, then the Performance tab, then Advanced, then Switch to: Memory
    b. Look at measurement Balloon for the hostname object.
    c. Is the value greater than 0?
        - Yes: Possible High Host Memory Demand. Go to step b to check for ballooning in the VMs
        - No: High Memory Demand is not present. Return to the Basic Troubleshooting flow.
2. Check for ballooning in the VM
    a. Select Chart Options, then Memory/Real-Time, then choose chart type Stacked Graph (Per VM).
    b. Look at measurement Balloon for all VM objects.
    c. Is the value greater than 0 for VMs experiencing performance problems?
        - Yes: High Host Memory Demand may be causing swapping within the Guest OS. Go to Memory-Related Performance Problems, for a discussion of possible solutions.

- No: High Memory Demand is not causing performance problems. Return to the Basic Troubleshooting flow.

Figure 16. Check for High Host Memory Demand

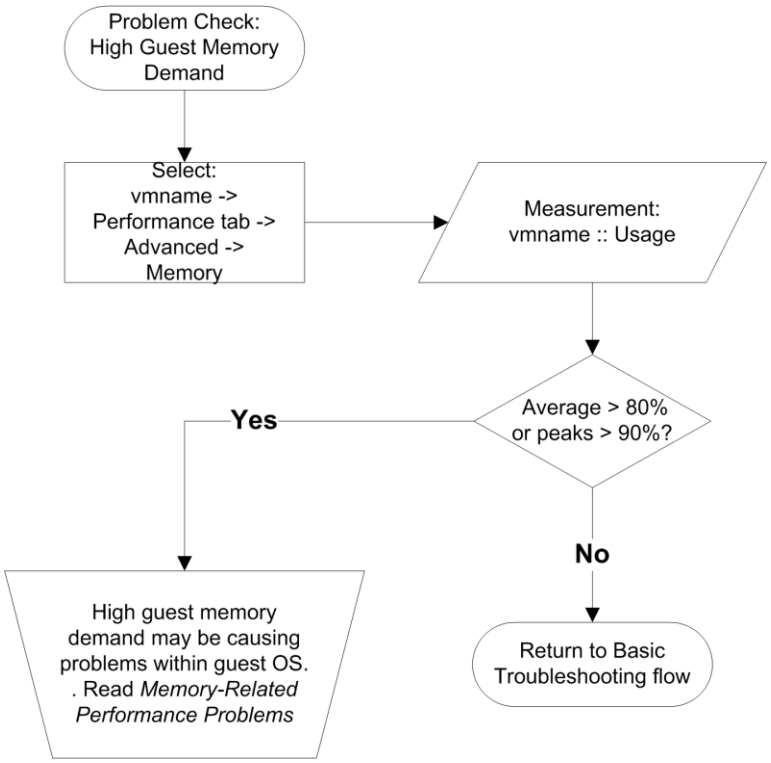


Check for High Guest Memory Demand

High demand for memory within a VM may indicate that memory-related problems are affecting the guest OS or application.

1. Check Memory Usage
    a. Select the VM, then the Performance tab, then Advanced, then Switch to: Memory
    b. Look at measurement Usage for the VM-name object.
    c. Is the average above 80% or are there peaks above 90%?
        - Yes: High guest memory demand may be causing performance problems. Go to Memory-Related Performance Problems, for a discussion of possible causes and solutions. If the performance problem is affecting the entire host. Repeat this check for other VMs on the host.
        - No: High guest memory demand is not present. Return to the Basic Troubleshooting flow.

Figure 17. Check for High Guest Memory Demand



Advanced Performance Troubleshooting with esxtop

Overview

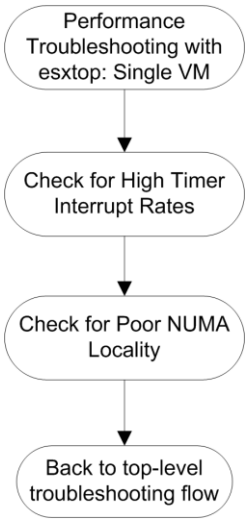
The basic troubleshooting flow in Basic Performance Troubleshooting for VMware ESX covers checks for the most common observable performance problems using the vSphere Client. However, there are some performance metrics, and therefore some performance problems, which are not observable though the vSphere Client. This section contains an advanced troubleshooting flow which uses esxtop to find observable problems that were not covered by the basic troubleshooting flow. All of the problems checked for using the vSphere Client in Basic Performance Troubleshooting for VMware ESX could also be checked for using data available in esxtop. However, we will not repeat those checks here.

The advanced troubleshooting flow is presented in Advanced Troubleshooting Flow. This flow gives a suggested order for checking for performance-related problems using esxtop. The checks for each of those problems are given in Advanced Problem Checks.

Advanced Troubleshooting Flow

The advanced troubleshooting flow is shown in Figure 18.

Figure 18. Advanced Performance Troubleshooting with esxtop



Advanced Problem Checks

The problem checks included in this section assume that you are running esxtop in the Service Console of the ESX host, or are connected to an ESX/ESXi host using resxtop.

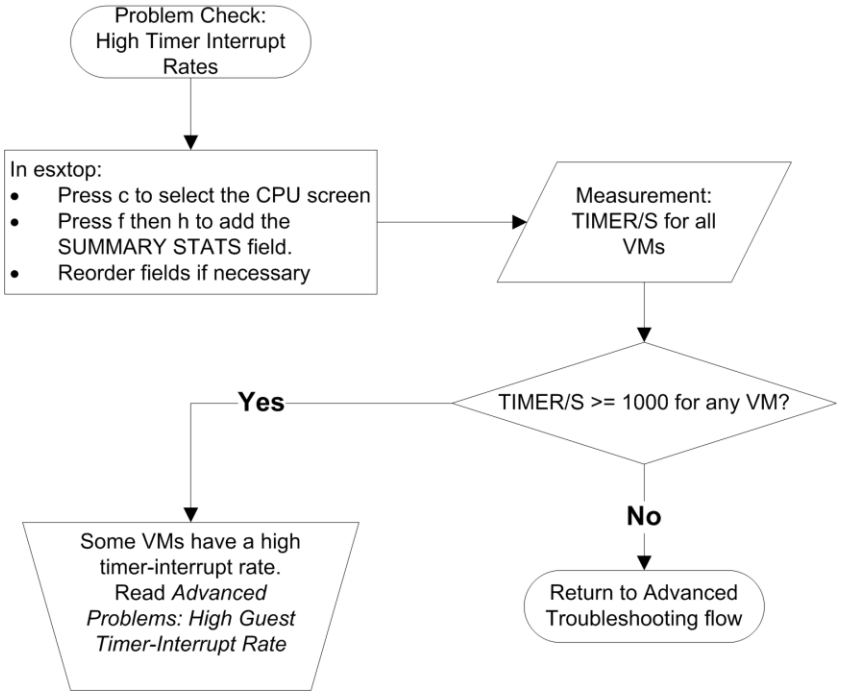
Check for High Timer-Interrupt Rates

A high timer-interrupt does not necessarily cause performance problems. However, it can add overhead that may impact the observed performance of an application running inside a VM.

1. Check Timer-Interrupt Rate
    a. Select the CPU screen by pressing c.

b. Add the SUMMARY STATS field by pressing f then h. Press any other key to return to the CPU screen.
c. Look at the TIMER/S measurement for all VMs. If this column is not visible, you will need to reorder the fields by pressing o and then pressing H a few times to move the SUMMARY STATS field over to the left. In ESX 4.0 you can press V to view only the VMs on this screen.
d. Is TIMER/S greater-than or equal to 1000 for any VM?
   - Yes: Some VMs have a high timer-interrupt rate. It may be possible to reduce this rate and thus reduce overhead. Go to High Guest Timer-Interrupt Rate for a discussion of possible causes and solutions.
   - No: The VMs on this host do not have a high timer-interrupt rate. Return to the Advanced Troubleshooting flow.

Figure 19. Check for High Timer-Interrupt Rates



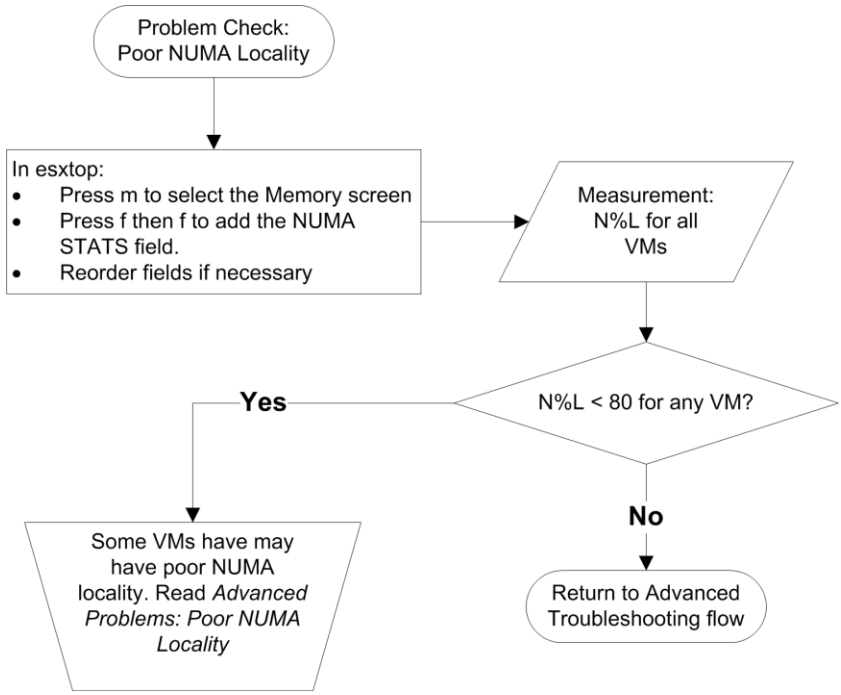
Check for Poor NUMA Locality

On a NUMA system, VMs are assigned to a home node, which represents a single physical processor chip. If the VM is using memory which is not local to the home node, performance may be degraded. However, there are performance tradeoffs involved in the assignment of home nodes and VM memory, and temporarily poor locality may cause less performance degradation than leaving a VM on a heavily-loaded home node. If this check shows poor NUMA locality on the ESX host, be sure to read the information in Poor NUMA Locality to determine whether this represents a real performance problem.

1. Check Memory Locality
   a. Select the Memory screen by pressing m.
   b. Add the NUMA STATS field by pressing f then f. Press any other key to return to the CPU screen.
   c. Look at the N%L column for all VMs. N%L is the percentage of a VM's memory that is located on its home node. If this column is not visible, you will need to reorder the fields by pressing o and then pressing F a few times to move the NUMA STATS field over to the left. In ESX 4.0 you can press V to view only the VMs on this screen.
   d. Is N%L less than 80% for any VM?
      - Yes: Some VMs may have poor NUMA locality. Go to Poor NUMA Locality, for a discussion of possible causes and solutions.

- No: The VMs on this host do not have a Poor NUMA Locality. Return to the Advanced Troubleshooting flow.

Figure 20. Check for Poor NUMA Locality



CPU-Related Performance Problems

Overview

CPU capacity is a finite resource. Even on a server which allows additional processors to be configured, there is always a maximum number of processors which can be installed. As a result, performance problems may occur when there are insufficient CPU resources to satisfy demand.

Excessive demand for CPU resources on an ESX host may occur for many reasons. In some cases the cause is straightforward. Populating an ESX host with too many VMs running compute-intensive applications can make it impossible to supply sufficient resources to any individual VM. However, sometimes the cause may be more subtle, related to inefficient use of available resources or non-optimal VM configuration.

In some cases, performance problems unrelated to excessive demand for CPU may be uncovered using CPU-related performance metrics. While we classify these as CPU-related problems, we will need to look elsewhere in the environment for root-causes and solutions. Low guest CPU usage due to high I/O response-times is an example of this type of problem.

The remainder of this section discusses the causes and solutions for specific CPU-related performance problems. The existence of these observable problems can be determined by using the associated problem-check diagrams in the troubleshooting sections of this document. The information in this section can be used to help determine the root-cause of the problem, and to identify the steps needed to remedy the identified cause.

Host CPU Saturation

Causes

The root-cause of host CPU saturation is simple. The VMs running on the host are demanding more CPU resource than the host has available. However, there are a few different scenarios in which this can occur, and for each scenario the approach to solving the problem may be slightly different.

The main scenarios for Host CPU Saturation are:

- The host has a small number of VMs, all with high CPU demand.
- The host has a large number of VMs, all with low to moderate CPU demand.
- The host has a mix of VMs with high and low CPU demand.

In the solutions to host CPU saturation presented in the following section, we discuss the appropriate approach to each scenario.

Solutions

Assuming that it is not possible to add additional compute resources to the host, there are four possible approaches to solving performance problems related to host CPU saturation.

- Reduce the number of VMs running on the host
- Increase the available CPU resources by adding the host to a DRS cluster
- Increase the efficiency with which VMs use CPU resources
- Use resource-controls to direct available resources to critical VMs

In this section we will discuss the implementation and implications of each of these solutions. The order that these solutions should be considered will vary depending on which of the previously discussed scenarios is present.

- When the host has a small number of VMs with high CPU demand, it may be preferable to attempt to solve the problem by first increasing the efficiency of CPU usage within the VM. VMs with high CPU demands are those most likely to benefit from simple performance tuning. In addition, unless care is taken with VM placement, moving VMs with high CPU demands to a new host may simply move the problem to that host.

- When the host has a large number of VMs with moderate CPU demands, reducing the number of VMs on the host by rebalancing load will be the simplest solution. Performance tuning on VMs with low CPU demands will yield smaller benefits, and will be time-consuming if the VMs are running different applications.

- When the host has a mix of VMs with high and low CPU demand, the appropriate solution will depend on available resources and skill sets. If additional ESX hosts are available, it may be best to rebalance load. If additional hosts are not available, or expertise exists for tuning the high-demand VMs, then increasing efficiency may be the best approach.

Reducing the number of VMs

The most straightforward solution to the problem of host CPU saturation is to reduce the demand for CPU resources by migrating VMs to ESX hosts with available CPU resources. In order to do this successfully, the performance charts available in the vSphere client should be used to determine the CPU usage of each VM on the host, and the available CPU resources on the target hosts. It is important to ensure that the migrated VMs will not cause CPU saturation on the target hosts. When manually rebalancing load in this manner, it is important to consider peak-usage periods as well as average CPU usage. This data is available in the historical performance charts when using vCenter to

manage ESX hosts. If vSphere's VMotion feature is configured, and the VMs and hosts meet the requirements for VMotion, then the load can be rebalanced with no downtime for the affected VMs.

If additional ESX hosts are not available, it is possible to eliminate host CPU saturation by powering off non-critical VMs. This will make additional CPU resources available for critical applications. Remember that even essentially idle VMs consume CPU resources. If powering-off VMs is not an option, then it will be necessary to increase efficiency or explore the use of resource controls.

Increasing CPU Resources with DRS Clusters

Using DRS clusters to increase available CPU resources is similar to the previous solution, in that additional CPU resources are made available to the set of VMs that have been running on the affected host. However, in a DRS Cluster, the rebalancing of load can be performed automatically, and it is not necessary to manually compute the load compatibility of specific VMs and hosts, or to account for peak-usage periods. Additional information about DRS Clusters is available in the *vSphere Resource Management Guide*.

Increasing VM Efficiency

The amount of CPU resources available on a given host is finite. In order to increase the amount of work that can be performed on a saturated host, it is necessary to increase the efficiency with which applications and VMs use those resources. CPU resources may be wasted due to sub-optimal tuning of applications and operating systems within VMs, or inefficient assignment of host resources to VMs.

The efficiency with which an application/OS combination uses CPU resources depends on many factors specific to the application, OS, and hardware. As a result, a discussion of tuning applications and operating-systems for efficient use of CPU resources is beyond the scope of this paper. Most application vendors provide performance tuning guides that document best-practices and procedures for their application. These guides often include OS-level tuning advice and best-practices. OS vendors also provide guides with performance tuning recommendations. In addition, there are numerous books and white-papers which discuss the general principles of performance tuning, as well as the application of those principles to specific applications and operating systems. These procedures, best-practices, and recommendations apply equally well to virtualized and non-virtualized environments. However, there are some application and OS-level tunings that are particularly effective in a virtualized environment. These are:

- Configure the application and guest OS to use large pages when allocating memory. See Using Large Memory Pages for details.

- Reduce the timer interrupt-rate for the guest OS. See Advanced Problem Checks for a problem check for high timer-interrupt rates.

A virtualized environment provides more direct control over the assignment of CPU and memory resources to applications than a non-virtualized environment. There are two key adjustments that can be made in VM configurations that may improve overall CPU efficiency. These are:

- Allocating more memory to a VM may enable the application running in the VM to operate more efficiently. Additional memory may enable the application to reduce I/O overhead or allocate more space for critical resources. Check the performance tuning information for the specific application to see if additional memory will improve efficiency. Remember that some applications need to be explicitly configured to use additional memory.

- Reducing the number of vCPUs allocated to VMs that are not using their full CPU resource will make more resources available to other VMs. Even if the extra vCPUs are idle, they still incur a cost in CPU resources, both in the VMware ESX scheduler, and in the guest OS overhead involved in managing the extra vCPU.

Using Resource-Controls

If it is not possible to rebalance CPU load or increase efficiency, or if all possible steps have been taken and host CPU saturation still exists, then it may be possible to reduce performance problems by using resource controls. Many applications, such as batch jobs or long-running numerical calculations, will respond to a lack of CPU resources by taking longer to complete, but will still produce correct and useful results. Other applications may experience failures, or may be unable to meet critical business requirements, when denied sufficient CPU resources. The resource controls available in vSphere 4 can be used to ensure that resource-sensitive applications always get sufficient CPU resources even when host CPU saturation exists. Additional information about resource controls is available in the vSphere Resource Management Guide.

Guest CPU Saturation

Causes

Guest CPU Saturation occurs when the application and OS running within a VM use all of the CPU resources that the ESX host is providing to that VM. The occurrence of guest CPU saturation does not necessarily indicate that a performance problem exists. Compute-intensive applications commonly use all available CPU resources, and even less intensive applications may experience periods of high CPU demand without experiencing performance problems. However, if a performance problem exists when guest CPU saturation is occurring, then steps should be taken to eliminate the condition.

Solutions

There are two possible approaches to solving performance problems related to guest CPU saturation.

- Increase the CPU resource provided to the application
- Increase the efficiency with which the VM uses CPU resources

Adding CPU resources is often the easiest choice, particularly in a virtualized environment. However, this approach will miss inefficient behavior in the guest application and OS that may be wasting CPU resources or, in the worst case, may be an indication of some erroneous condition within the guest. If a VM continues to experience CPU saturation even after adding CPU resources, the tuning and behavior of the application and OS should be investigated.

Increasing CPU Resources

In a vSphere environment, the following options are available for increasing the CPU resources available to a VM:

- Add additional vCPUs to the virtual machine. This is possible as long as the VM is not already configured with the maximum allowable number of vCPUs.

- Migrate the VM to an ESX host running on a server with more powerful processors. Published results on the VMmark benchmark can be used as a rough guide to the relative performance of different hardware platforms. However, different applications will receive different levels of performance benefit from specific hardware features.

- If allowed by the application, add additional VMs running same application, and balance the workload over all of the VMs. Depending on the application, the VMs may be added on the same or different ESX hosts.

Increasing VM Efficiency

For a discussion of increasing VM efficiency, see the Solutions section under Host CPU Saturation.

Using only one vCPU in an SMP VM

Causes

When a VM configured with more than one vCPU is only actively using one of those vCPUs, resources which could be used to perform useful work are being wasted. The possible causes for CPU utilization by a vSMP VM only one vCPU are:

- **Guest OS configured with a uni-processor kernel (Linux) or Hardware Abstraction Layer (HAL) (Windows).** In order for a VM to take advantage of multiple vCPUs, the guest OS running in the VM must be able to recognize and use multiple processor cores. If the VM is doing all of its work on vCPU0, then the guest OS may be configured with a kernel or HAL that can only recognize a single processor core. Follow the documentation provided by your OS vendor to check for the type of kernel or HAL being used by the guest OS.

- **Application is pinned to a single core in the guest OS.** Many modern OSes provide controls for restricting applications to run on only a subset of available processor cores. If these controls have been applied within the guest OS, then the application may run on only vCPU0. To determine whether this is the case, inspect the commands used to launch the application, or other OS-level tunings applied to the application. This must be accompanied by an understanding of the OS vendor's documentation regarding restricting CPU resources.

- **Application is single-threaded.** Many applications, particularly older applications, are written with only a single thread of control. These applications cannot take advantage of more than one processor core. Note however that many guest OSes will spread the run-time of even a single-threaded application over multiple processors. Thus running a single-threaded application on a vSMP VM may lead to low guest CPU utilization, rather than utilization on only vCPU0. Without knowledge of the application design, it may be difficult to determine whether an application is single-threaded other than by observing the behavior of the application in a vSMP VM. If, even when driven at peak load, the application is unable to achieve CPU utilization of more than 100% of a single CPU, then it is likely that the application is single threaded. However, poor application or guest OS tuning, or high I/O response-times, could also lead to artificial limits on the achievable utilization.

Solutions

In this section we discuss solutions for each of the root-causes of an SMP VM utilizing only one vCPU.

- **Guest OS configured with a uni-processor kernel (Linux) or Hardware Abstraction Layer (HAL) (Windows).** If the check of the guest OS showed that it is using a uni-processor kernel or HAL, follow the OS vendor's documentation for upgrading to an SMP kernel or HAL.

- **Application is pinned to a single core in the guest OS.** If the application was pinned to a single core in the guest OS, then either the OS-level controls should be removed, or, if the controls are determined to be appropriate for the application, the number of vCPUs should be reduced to match the number that will actually be used by the VM.

- **Application is single-threaded.** If the VM is running only one application, and that application is single-threaded, then the VM will be unable to take advantage of more than one vCPU. The number of vCPUs allocated to the VM should be reduced to one.

Low Guest CPU Utilization

Causes

Low CPU utilization in a VM is generally not a problem. In normal cases, it simply indicates that the application running in the VM is experiencing low demand. However, if the application is experiencing performance problems, such as unacceptably long response-times, low CPU utilization may be an indicator of the underlying root-cause.

The following may cause an application to experience performance problems even when CPU utilization is low:

- **High storage response-times.** Time spent waiting for storage accesses to complete can lead to high response-times, or other performance problems, even when CPU utilization is low.

- **High response-times from external systems.** Applications which form part of a multi-tier solution will often need to wait for responses to external requests before completing an operation. One example of this is an application server which requests data from a database server running in a separate VM or on a separate system. If the external system is slow, the performance observed on the local application will be poor, even though the CPU utilization is low. Refer to performance monitoring information for the specific application in order to determine whether this is occurring.

- **Poor application or OS tuning.** Many applications allocate a finite number of certain resources or structures, such as worker threads or database connections, for use when handling requests or performing operations. If an inadequate number of these resources are allocated, the application may exhibit poor performance even though CPU utilization is low. Refer to performance monitoring information for the specific application in order to determine whether this is occurring.

- **Application is pinned to cores in the guest OS.** Many modern OSes provide controls for restricting applications to run on only a subset of available processor cores. If these controls have been applied within the guest OS, then the application may run on only some available cores of a vSMP VM, leading to low overall CPU utilization.

- **Too many vCPUs configured.** Many applications are designed in a manner that prevents them from taking advantage of large numbers of CPUs. If a VM is configured with more vCPUs than it needs or can use, the overhead of managing those vCPUs may lead to performance problems. This overhead may be in the application, the guest OS, or in VMware ESX.

- **Restrictive resource allocations.** vSphere provides controls which allow the CPU and Memory resources available to a VM to be limited. If the resource limits for a VM are set too low, performance problems may occur even though the VM appears to have low CPU utilization.

Some of these causes are relatively easy to check for, while others require application-specific performance monitoring and tuning. Here we recommend a procedure for determining the root-cause based on problem likelihood and complexity of problem checks.

1. **Check for High storage response-times.** Refer to the problem checks for Slow and Overloaded Storage to determine whether this problem exists. If so, address the problem and recheck application performance.

2. **Check for restrictive resource allocations.** Examine the resource settings for the VM and for its parent resource pools to determine whether there are limits set that would prevent the VM from obtaining necessary resources. One way to check for restrictive CPU allocations is to use in-guest performance monitoring tools to observe the CPU utilization reported by the guest OS. If the in-guest tools report CPU utilizations near 100% when the vSphere Client is reporting low CPU utilization, then a CPU limit has likely been placed on the VM. Check the resource allocations for the VM and its resource pool.

3. **If the VM is configured with multiple vCPUs, reduce the number of vCPUs and recheck performance.** Be sure to consider peak loads when determining the appropriate number of vCPUs.

4. **Check for high external response-times and poor application/OS tuning.** The order in which these factors should be checked will depend on knowledge of the application and environment which is beyond the scope of this document. Refer to application-specific guidance for more details.

Solutions

In this section we discuss solutions for each of the root-causes for Low Guest CPU Utilization.

- **High storage response-times.** If high storage response-times are occurring, refer to Storage-Related Performance Problems for possible causes and solutions.
- **High response-times from external systems.** Refer to performance tuning information for the specific applications in order to determine appropriate solutions.
- **Poor application or OS tuning.** Refer to performance tuning information for the specific applications and OSes in order to determine appropriate solutions.
- **Application is pinned to cores in the guest OS.** If the application was pinned to a subset of available cores in the guest OS, then either the OS-level controls should be removed, or, if the controls are determined to be appropriate for the application, the number of vCPUs should be reduced to match the number that will actually be used by the VM.
- **Too many vCPUs configured.** Reduce the number of vCPUs configured for the VM.
- **Restrictive resource allocations.** Remove or increase the limits on the resources available to the VMs.

High Utilization on pCPU0

Causes

This problem refers to situations where the CPU utilization on physical CPU core 0 (pCPU0) is disproportionately high compared to the utilization on other CPU cores. If your system is experiencing CPU utilization that is uniformly high on all CPU cores, refer to the problem check for Host CPU Saturation and the associated cause and solution sections.

In VMware ESX, the Service Console is restricted to running only on pCPU0. In many vSphere deployments, management agents from various vendors are run inside the Service Console. When an agent demands a large amount of CPU resources, or many less demanding agents are run, the utilization on pCPU0 can rise out of proportion to the overall CPU utilization. Note that this does not apply to the ESXi edition, which does not have a Service Console.

High utilization on pCPU0 may impact the performance of other VMs running on the ESX host. The ESX scheduler will attempt to run VMs on other processors. However, high CPU utilization by the Service Console decreases resources available to VMs. In particular, vSMP VMs running on NUMA systems may experience performance impacts when assigned to the home node that includes pCPU0.

Solutions

In order to reduce the utilization of pCPU0 due to agents running in the service console, it may be necessary to reduce the number of agents, or reduce the amount of work being performed by the agents. Refer to the vendor of your agents for additional information. Also, ensure that all agents are up-to-date. Vendors occasionally release updated versions to address performance-related issues.

Memory-Related Performance Problems

Overview

Host memory, like CPU capacity, is a limited resource. However, VMware ESX incorporates sophisticated mechanisms that maximize the use of available memory through sharing and resource-allocation controls. These mechanisms allow more memory to be allocated to virtual machines than is physically configured on the system. VMware ESX attempts to support this over-commitment of memory while providing sufficient resources for each VM to operate with peak performance. However, in cases where the VMs on an ESX host are, in the aggregate, actually using more memory than is available, ESX must take steps to ensure correct and reliable operation of the VMs. The cost of keeping the VMs operating correctly is sometimes a decrease in the performance of one or more VMs.

The measures that VMware ESX uses to maintain correctness when memory resources are over-committed are:

- **Ballooning:** When VMware tools are installed in a guest OS, a device-driver called the memory balloon driver is installed. When VMware ESX needs additional memory, it uses the balloon driver to reclaim memory from the VM. Ballooning is part of normal operation when memory is over-committed, and the fact that ballooning is occurring is not necessarily an indication of a performance problem. The use of the balloon driver enables the guests to give up memory pages that they are not currently using. However, if ballooning causes the guest to give up memory that it actually needs, performance problems can occur due to guest OS paging.

- **Swapping:** When VMware ESX needs additional memory, but has reclaimed as much as possible using ballooning, it must resort to swapping out portions of VM memory to swap files created on disk. Swapping by ESX is a measure of last resort which is taken to maintain correct operation of the VMs running on that host. However, since swapping moves VM memory to disk without regard to the portions that are actually being used, it can cause serious performance problems. Note that ESX swapping is distinct from swapping performed by a guest OS due to memory pressure within a VM. Guest OS-level swapping may occur even when the ESX host has ample memory resources (see High Guest Memory Demand).

Additional information about the memory-management mechanisms in VMware ESX can be found in the *vSphere Resource Management Guide*.

In order to understand the causes of memory-related performance problems, it is useful to understand the following terms:

- **Memory Overcommitment:** Host memory is over-committed when the total memory space allocated to powered-on VMs, including overhead, is greater than the amount of memory physically available in the host. To determine whether memory is over-committed on a host:

    1. In the vSphere client, select the host, and then the Summary tab. The value under Capacity next to the Memory Usage bar is the total amount of memory available in the host.

    2. In the vSphere client, select the host, then the Virtual Machines tab.

    3. Sort the displayed VMs by the State column, so that all VMs that are Powered On are displayed together.

    4. Add the values in the Memory Size column for all VMs in the Powered On state. If the Memory Size column is not displayed, right-click on the table header, and select the Memory Size column. The result is the total memory explicitly allocated to powered-on VMs.

    5. In the vSphere Client, select the host, then the Performance tab, than Advanced, then Switch To: Memory. Change chart options to include the Overhead metric in the display.

6. If the total Memory Size of all powered-on VMs plus the Overhead is greater than the capacity determined in step a, then host memory is over-committed.

- **Memory Overhead:** When a VM is powered on, VMware ESX reserves memory for virtualization data-structures required to support that VM. This overhead memory is considered reserved, and is not available for ballooning or swapping. The size of the memory overhead for each VM can be found by selecting the VM in the vSphere Client and looking in the General section of the Summary tab. For more information on overhead memory, including per-VM overhead size based on VM memory size and number of vCPUs, see the *vSphere Resource Management Guide*.

- **Memory Reservation:** A memory reservation is a lower-bound placed on the amount of memory that ESX will make available to a VM. A VM will always have access to the memory reserved for it, and ESX will not attempt to reclaim this memory through ballooning or swapping. To determine the total amount of memory reserved for a VM:

    1. In the vSphere client, select the host, then the Resource Allocation tab.

    2. The value labeled Reserved Capacity under Memory is the total current memory reservation on that host. This value includes both memory reserved for VMs, and the total memory overhead.

- **Total Balloonable Memory:** For each VM, there is a maximum amount of memory that ESX will reclaim through ballooning. ESX sets a maximum balloon size based on the memory size of the VM and the type of guest OS. However, the amount ESX can actually reclaim from a VM using ballooning may be limited by the VM's memory reservation. In addition, ESX will be unable to reclaim memory using ballooning if the balloon driver is not running inside the VM. The sum of the maximum amount of memory actually balloonable for each VM is the total balloonable memory. This is the maximum amount of memory that ESX can reclaim through ballooning before it must resort to swapping.

- **Shared Memory:** VMware ESX uses memory sharing to allow multiple VMs to share memory pages which have identical contents. This reduces the total amount of physical memory required to support a number of VMs. To determine the current memory savings due to memory sharing:

    1. In the vSphere client, select the host, then the Performance tab, then Switch To: Memory.

    2. Look at measurements Shared and Shared Common.

    3. The total memory savings due to memory sharing is Shared - Shared Common.

- **Total Active Memory:** Over any period in time, a VM may only use a portion of the memory it has been allocated. The amount of memory that it has used in the recent past is its active memory. The sum of the active memory of all VMs is the total active memory for the host.

The remainder of this section discusses the causes and solutions for specific memory-related performance problems. The existence of these observable problems can be determined by using the associated problem-check diagrams in the troubleshooting sections of this document. The information in this section can be used to help determine the root-cause of the problem, and to identify the steps needed to remedy the identified cause.

Active VM Memory Swapping

Causes

VMware ESX uses swapping as a last resort to maintain correct operation of VMs when there is not sufficient memory available to meet the needs of all VMs simultaneously. Swapping almost always causes performance problems with the VM whose memory is being swapped. It can also cause performance problems for other VMs due to the added overhead of managing swapping.

The basic cause of memory swapping is memory over-commitment using VMs with high memory demands. As a rough approximation, memory swapping will occur when:
Total_active_memory > (Memory_Capacity - Memory_Overhead) + Total_balloonable_memory + Page_sharing_savings

In other words, when the total amount of active memory exceeds the memory capacity of the host minus the VM overhead, even accounting for the benefits of ballooning and page sharing, the host will be forced to resort to swapping. Based on this, we see that the following conditions can cause memory swapping:

- **Excessive memory over-commit.** The level of memory over-commitment is too high for the combination of memory capacity, VM allocations, and VM workload. In addition to these factors, the amount of overhead memory must be taken into consideration when using memory over-commit.

- **Memory over-commit with memory reservations.** The total amount of balloonable memory may be reduced by memory reservations. If a large portion of a host's memory is reserved for some VMs, the host may be forced to swap the memory of other VMs. This may happen even though VMs are not actively using all of their reserved memory.

- **Balloon drivers not running or disabled.** If the balloon driver is not running, or has been deliberately disabled in some VMs, the amount of balloonable memory on the host will be decreased. As a result, the host may be forced to swap even though there is idle memory that could have been reclaimed through ballooning.

  In rare cases when an error occurred during VMware tools installation, the vSphere Client may report the tools status as OK even though the balloon driver is not running. The easiest way to determine whether the balloon driver for a VM is not running is through esxtop.

  - Open esxtop or resxtop for the host.
  - Switch to the Memory screen, and add the MCTL fields.
  - If the MCTL? field contains N for any VM, then the balloon driver is not running in that VM.

Note: The balloon driver is an important part of the proper operation of the ESX memory management system. The balloon driver should never be deliberately disabled in a VM. This may cause unintended performance problems (e.g. swapping), and makes tracking down memory-related problems difficult. vSphere provides other mechanisms, such as memory reservations, for controlling the amount of memory available to a VM.

Solutions

There are four solutions to performance problems caused by VM Memory Swapping. They are:

- Reduce the level of memory over-commit.
- Enable the balloon driver in all VMs.
- Reduce memory reservations.
- Use resource controls to dedicate memory to critical VMs.

In this section we will discuss the implementation and implications of each of these solutions.

Reduce the level of memory over-commit

In most situations, reducing memory over-commit levels is the proper approach for eliminating swapping on an ESX host. However, be sure to consider all of the factors discussed in the previous sections to ensure that the reduction is adequate to eliminate swapping. If other approaches are used, be sure to monitor the host to ensure that swapping

has been eliminated. In all cases, ensure that the balloon driver is running in all VMs.

The follow steps can be taken to reduce the level of memory over-commit.

- **Add physical memory to the ESX host.** Adding physical memory to the host will reduce the level of memory over-commit, and may eliminate the memory pressure that caused swapping to occur.

- **Reduce the number of VMs running on the ESX host.** The level of memory over-commit can be reduced by migrating VMs to ESX hosts with available memory resources. In order to do this successfully, the performance charts available in the vSphere client should be used to determine the memory usage of each VM on the host, and the available memory resources on the target hosts. It is important to ensure that the migrated VMs will not cause swapping to occur on the target hosts. If vSphere's VMotion feature is configured, and the VMs and hosts meet the requirements for VMotion, then the load can be rebalanced with no downtime for the effected VMs.

  If additional ESX hosts are not available, it is possible to reduce the number of VMs by powering off non-critical VMs. This will make additional memory resources available for critical applications. Remember that even essentially idle VMs consume some memory resources.

- **Increase available memory resources by adding the host to a DRS cluster.** Using DRS clusters to increase available memory resources is similar to the previous solution, in that additional memory resources are made available to the set of VMs that have been running on the affected host. However, in a DRS Cluster, the rebalancing of load can be performed automatically, and it is not necessary to manually compute the compatibility of specific VMs and hosts, or to account for peak-usage periods. Additional information about DRS Clusters is available in the vSphere Resource Management Guide.

- **Maximize page sharing.** Consolidating VMs with same OS on same server can maximize the benefit of page sharing, making more memory available to VMs. However, this is an unreliable solution to swapping, as the amount of memory saved through page sharing will depend on the workloads running in the VMs, and may change over time.

Enable balloon driver in all VMs

In order to maximize the ability of ESX to recover idle memory from VMs, the balloon driver should be enabled in all VMs. This should be done regardless of which other solutions are used. If a VM has critical memory needs, then reservations and other resource controls should be used to ensure that those needs are satisfied.

In cases of excessive memory over-commit, enabling the balloon drivers will only delay the onset of swapping, and is not a sufficient solution. The level of memory over-commit should be reduced as well.

Reduce memory reservations

If large memory reservations are causing the ESX host to swap memory from VMs without reservations, then the need for those reservations should be re-evaluated. If a VM with reservations is not using its full reservation, even at peak periods, then the reservation should be reduced. If the reservations cannot be reduced, either because the memory is needed or for business reasons, then the level of memory over-commit must be reduced.

Use resource controls to dedicate memory to critical VMs

As a last resort, when it is not possible to reduce the level of memory over-commit, it is possible to use memory reservations to prevent swapping of memory from performance critical VMs. However, this will only move the swapping problem to other VMs, whose performance will be severely degraded. In addition, swapping of other VMs may still impact the performance of VMs with reservations due to added disk traffic and memory-management overhead.

Past VM Memory Swapping

Causes

In some cases, VM memory can remain swapped to disk even though the ESX host is not actively swapping. This will occur when high memory activity caused some VM memory to be swapped to disk (see the "Solutions" section under Active VM Memory Swapping), and the VM whose memory was swapped has not yet attempted to access the swapped memory. As soon as the VM does access this memory, the ESX will swap it back in from disk. Whether this causes additional swapping activity would depend on the state of host memory at the time that the swapped memory is accessed.

Having VM memory which is not being actively used swapped out to disk will not cause performance problems. However, it may be an indication of the cause of performance problems which were observed in the past (that is, due to active swapping), and it may indicate that memory-related performance problems may recur at some point in the future.

There is one common situation that can lead to VM memory being left in swap even though there was no observed performance problem. When a VM's guest OS first starts, there will be a period of time before the memory-balloon driver begins running. In that time, the VM may access a large portion of its allocated memory. For example, some guest OSes will initialize their entire memory space when booting. If many VMs are powered-on at the same time, the spike in memory demand together with the lack of running balloon drivers may force ESX to resort to swapping. Some of the memory that is swapped may never again be accessed by a VM, causing it to be remain swapped to disk even though there is adequate free memory. This type of swapping may slow the boot process, but will not cause problems once the OSes have finished booting.

Solutions

The first step in addressing past VM memory swapping is to determine whether the memory was swapped due to many VMs being powered-on simultaneously. Check the logs from the guest OSes of the VMs on the host to determine whether many VMs were booted at close to the same time. If so, this is the likely cause of the swapping. Otherwise, refer to the "Solutions" section under VM Memory Swapping for solutions to VM memory swapping.

High Host Memory Demand

Causes

High host memory demand will not necessarily cause performance problems. As long as memory is not over-committed, and sufficient memory capacity is available for VM overhead and for VMware ESX internal operation, all of host memory can be used without impact on VM performance. Even if memory is over-committed, ESX can use memory ballooning to reclaim unused memory from VMs to meet the needs of more demanding VMs. As long as memory is not so over-committed that swapping occurs (see the problem check for VM Memory Swapping), ballooning can maintain good performance in most cases. However, if overall memory demand is high, the ballooning mechanism may reclaim memory that is actually needed by an application or guest OS. In addition, some applications are highly sensitive to having any memory reclaimed by the balloon driver.

In order to determine whether ballooning is affecting performance of a VM, it is necessary to investigate the use of memory within the VM. If a VM with a balloon value of greater than zero, as identified in the problem check for High Host Memory Demand, is experiencing performance problems, the next step is to examine paging and swapping statistics from within the guest OS. In Windows guests, these statistics are available through perfmon, while in Linux guests they are provided by tools such as vmstat or sar. Refer to performance-monitoring documentation for your guest OS to determine how to look for excessive paging and swapping activity. One important thing to remember is that the values provided by these performance monitoring tools within the guest OS are not strictly accurate. However, they will provide enough information to determine whether the VM is experiencing memory pressure.

Solutions

If ballooning of guest memory due to high host memory usage is causing performance problems in the guest, there are two possible approaches to addressing this problem.

- **Use resource-controls to direct available resources to critical VMs.** In order to prevent the host from ballooning memory from the VM, memory reservations can be set for the VM to ensure that it has adequate memory available. Monitor the actual amount of memory used by the VM to determine the appropriate reservations value.

  Note that increasing the reservation for the VM may cause the ESX host balloon memory from other VMs. In some cases it may also lead to swapping of VM memory (see the previous section). Hosts with memory over-commit should be monitored for swapping.

- **Eliminate memory over-commit on the host.** Memory ballooning can be eliminated on the host by eliminating memory over-commit. This can be done using the same approaches discussed for reducing excessive memory over-commit in Active VM Memory Swapping. However, this approach will prevent ESX from taking advantage of otherwise idle memory. The overall level of memory usage should be considered before using this approach to solve the problem of guest OS swapping in one VM.

Disabling the balloon driver in the VM is never the correct solution. Instead, use resource controls to maintain memory availability for the VM.

High Guest Memory Demand

Causes

Performance problems can occur when the application and guest OS running in a VM are using a high percentage of the memory allocated to them. High memory demand within a guest can lead to swapping or paging by the guest OS, as well as other, application-specific, performance problems.

The causes of high memory demand will vary depending on the application and guest OS being used. The high Memory %-Used value reported in the vSphere client for the VM is just an indicator of what is happening in the guest. It is necessary to use application and OS specific monitoring and tuning documentation in order to determine why the guest is using most of its allocated memory, and whether this is causing performance problems.

Solutions

If it is determined that the high demand for memory is causing performance problems in the guest, the following steps can be taken to address the performance problems.

- **Configure additional memory for the VM.** In a vSphere environment, it is much easier to add memory to a guest than would be possible in a physical environment. Before configuring additional memory, ensure that the guest OS and application used can take advantage of the added memory. For example, on 32-bit OSes, some OSes and most applications can only access 4GB of memory, regardless of how much is configured. Also, it is important to check that the new allocation will not cause excessive over-commit on the host that will lead to swapping.

- **Tune the application to reduce its memory demand.** The methods for doing this are application specific.

Storage-Related Performance Problems

Overview

The performance of the storage infrastructure can have a significant impact on the performance of applications, whether running in a virtualized or non-virtualized environment. In fact, problems with storage performance are the most common causes of application-level performance problems.

Unlike CPU or memory resources, storage performance capacity is not strictly limited by server architecture. Storage networking technologies, such as Fibre Channel, NFS, and iSCSI, allow storage performance capacity, in terms of both bandwidth and I/O Operations-Per-Second (IOPS), to be expanded well beyond the needs of most applications. However, storage sizing and configuration is a complex task, with a large number of inter-related variables. For example, the number of disks in a LUN, RAID level, the size of caches in storage adapters or arrays, the capacity and configuration of storage-network links, and the sharing of disks and LUNs among multiple applications, are just some of the variables that can impact storage performance.

VMware vSphere is capable of achieving high performance with local disks, or with any supported storage-networking technology. However, just as in a non-virtualized environment, this performance depends on proper configuration of the storage subsystem for the application. In some cases, applications that were consolidated on vSphere from a physical environment end up with fewer disk resources. In other cases, sharing of storage resources among previously isolated applications can lead to performance problems. In general, the causes of, and solutions to, storage performance problems are identical in non-virtualized and virtualized environments.

Overloaded Storage

Causes

When storage is severely overloaded, operation time-outs can cause commands already issues to disk to be aborted. This can lead to serious application-level performance problems. If a storage device is experiencing command aborts, the cause must be identified and corrected.

There are two main causes of overloaded storage.

- **Excessive demand being placed on the storage device.** The storage load being placed on the device exceeds the ability of the device to respond.

- **Misconfigured storage.** Storage devices have a large number of configuration parameter, such as number of disks per LUN, the RAID level of a LUN, and the assignment of array cache to a LUN. Choices made for any of these variables can affect the ability of the device to handle the I/O load. Storage vendors typically provide guidelines on proper configuration and expected performance for different levels and types of I/O load.

Solutions

Due to the complexity and variety of storage infrastructures, it is impossible to specify specific solutions for slow or overloaded storage in this document. vSphere is capable of high storage performance using any supported storage technology, and, beyond rebalancing load, there is typically little that can be done from within vSphere to solve problems related to slow or overloaded storage. Documentation from your storage vendor should be followed to monitor the demand being placed on the storage device, and vendor-specific configuration recommendations should be followed to configure the device for the demand. If the device is not capable of satisfying the I/O demand with good performance, the load should be distributed among multiple devices, or faster storage should be obtained.

To help guide the investigation of performance issues in the storage infrastructure, we offer some general notes on storage performance, and the differences between physical and virtual environments, that should be considered when investigating storage performance problems in a vSphere environment.

- **Size storage for performance as well as capacity.** As physical disks continue to grow in storage capacity, it becomes possible to satisfy the storage capacity requirements of applications with a very small number of disks. However, the I/O rate that each individual disk can handle is limited. As a result, it is possible to fit the storage for an application on a disk device that cannot physically handle the I/O rates that it can generate. The ability of a storage device to handle the bandwidth and IOPS demands of an application is as important a selection criterion as is the capacity of the device. Storage vendors typically provide data on the IOPS that their devices can handle for different types of I/O loads and configuration options.

  Sizing storage for performance is particularly important in a virtualized environment. When applications are running on a server in a non-virtualized environment, the storage for that application is often placed on a dedicated LUN with specific performance characteristics. When moved to a vSphere environment, the storage for an application may be taken from a pre-existing VMFS volume, which may be shared by multiple VMs running different applications. If the LUN on which the VMs are placed was not sized and configured to meet the needs of all VMs, the storage performance available to an application, and therefore its overall performance, may suffer. As a result it is critical to consider the performance capabilities of VMFS volumes, and their backing LUNs, as well as available capacity, when placing VMs.

- **Moving to a virtualized environment changes storage workloads.** In a non-virtualized environment, the configuration of LUNs is often based on the I/O characteristics of single applications. Characteristics such as IOPS rate, I/O size, and disk access-pattern, all affect the proper configuration of storage to satisfy the performance demands of an application. When multiple applications are consolidated onto a single VMFS volume, the workload presented to that volume will be different than the storage workload of the individual applications. A particular example is that the interleaving of I/Os from multiple applications may cause sequential request streams to be presented to the storage device as random requests. This will increase the number of physical disk devices needed to meet the storage performance needs of the applications.

- **Understand the load being placed on storage devices.** In order to properly troubleshoot storage performance, it is important to understand the load being placed on the storage device. Many storage arrays have tools that allow workload statistics to be captured. In addition to providing information on IOPS and I/O sizes, these tools may provide information on queuing in the array, and statistics regarding the effectiveness of the caches in the array. Refer to the vendor's documentation for details.

  VMware ESX also provides tools for understanding storage workloads. In addition to the high-level data provided by the vSphere Client and esxtop, the vscsiStats tool can provide detailed information on the I/O workload generated by a VM's virtual SCSI device. See http://communities.vmware.com/docs/DOC-10095 for more details.

- **Simple benchmarks can help isolate storage performance problems.** Once slow storage has been identified, it can be difficult to troubleshoot the problem using application-level loads. When using live applications, problems may occur intermittently or only during peak periods. Driving applications with synthetic loads can be time-consuming and require complex tools. Disk benchmarking tools, such as IOmeter, can be used to generate loads similar to application loads in a controllable manner. See http://communities.vmware.com/docs/DOC-3961 for more information.

- **Consider the tradeoff between memory capacity and storage demand.** Some applications can take advantage of additional memory capacity to cache frequently used data, thus reducing storage loads. This typically has the added benefit of improving application performance. Because running applications in VMware ESX makes it easy to add additional memory capacity to a VM, this tradeoff should be considered when storage is unable to meet existing demands. Refer to the application-specific documentation to determine whether an application can take advantage of additional memory. Some applications require changes to configuration parameters in order to make use of additional memory.

Slow Storage

Causes

Slow storage is the most common cause of performance problems in a vSphere environment. The Physical Device Read Latency and Physical Device Write Latency performance metrics provided by the vSphere Client show the average time for an I/O operation to complete from the time it is submitted to the hardware adapter until the completion notification is provided back to ESX. As a result, these values represent performance achieved by the physical storage infrastructure. The values for these metrics specified in the problem check for Slow Storage represent response-times beyond which performance problems may exist in the storage infrastructure. However, in order to understand whether these values represent an actual problem, it is necessary to understand the storage workload.

There are three main workload factors that affect the response-time of a storage subsystem:

- **I/O arrival-rate:** A given configuration of a storage device will have a maximum rate at which it can handle specific mixes of I/O requests. When bursts of requests exceed this rate, they may need to be queued in buffers along the I/O path. This queuing can add to the overall response-time.

- **I/O Size:** The transmission rate of storage interconnects, and the speed at which an I/O can be read from or written to disk, are fixed quantities. As a result, large I/O operations will naturally take longer to complete. A response-time that is slow for small transfers may be expected for larger operations.

- **I/O Locality:** Successive I/O requests to data that is stored sequentially on disk can be completed more quickly than those that are spread randomly. In addition, read requests to sequential data are more likely to be completed out of high-speed caches in the disks or arrays.

Storage devices typically provide monitoring tools that allow data to be collected in order to characterize storage workloads according to these, and other, factors. In addition, storage vendors typically provide data on recommended configurations and expected performance of their devices based on these workload characteristics. If the problem check for Slow Storage indicated that storage may be causing performance problems, the monitoring tools should be used to collect workload data, and the storage response-times should be compared to expectations for that workload. If investigation determines that storage response-times are unexpectedly high, corrective action should be taken.

Solutions

The solutions for Slow Storage are the same as those discussed for Overloaded Storage. See the "Solutions" section under Overloaded Storage (above) for details.

Network-Related Performance Problems

Overview

The networking performance that can be achieved by an application depends on many factors. These factors can affect not only network-related performance metrics, such as bandwidth or end-to-end latency, but also metrics such as CPU overhead and achieved performance of applications. Among these factors are the network protocol, guest OS network stack and tuning, NIC capabilities and offload features, CPU resources, and link bandwidth. In addition, less obvious factors such as congestion due to other network traffic and buffering along the source-destination route can lead to network-related performance problems. These issues are identical in virtualized and non-virtualized environments.

Networking in a virtualized environment does add new factors that must be considered when troubleshooting

performance problems. The use of virtual switches to combine the network traffic of multiple VMs onto a shared set of physical uplinks places greater demands on a host's physical NICs, and the associated network infrastructure, than in most non-virtualized environments.

The remainder of this section discusses the causes and solutions for specific Network-related performance problems. The existence of these observable problems can be determined by using the associated problem-check diagrams in the troubleshooting sections of this document. The information in this section can be used to help determine the root-cause of the problem, and to identify the steps needed to remedy the identified cause.

Dropped Receive Packets

Causes

Network packets may get stored (buffered) in queues at multiple points along their route from the source to the destination. Network switches, physical NICs, device drivers, and network stacks may all contain queues where packet data or headers are buffered before being passed to the next step in the delivery process. These queues are finite in size. When they fill up, no more packets can be received at that point on the route, and additional arriving packets must be dropped. TCP/IP networks use congestion-control algorithms that limit, but do not eliminate, dropped packets. When a packet is dropped, TCP/IP's recovery mechanisms work to maintain in-order delivery of packets to applications. However these mechanisms operate at a cost to both networking performance and CPU overhead, a penalty that becomes more severe as the physical network speed increases.

vSphere presents virtual network-interface devices, such as the vmxnet or virtual e1000 devices, to the guest OS running in a VM. For received packets, the virtual NIC buffers packet data coming from a virtual switch (vSwitch) until it is retrieved by the device-driver running in the guest OS. The vSwitch, in turn, contains queues for packets sent to the virtual NIC.

If the Guest OS does not retrieve packets from the virtual NIC rapidly enough, the queues in the virtual NIC device can fill up. This can in turn cause the queues within the corresponding vSwitch port to fill. If a vSwitch port receives a packet bound for a VM when its packet queue is full, it must drop the packet. The count of these dropped packets over the selected measurement interval is available in the performance metrics visible from the vSphere client. In previous versions of VMware ESX, this data was only available from esxtop.

The following problems can cause the guest OS to fail to retrieve packets quickly enough from the virtual NIC:

- **High CPU utilization.** When the applications and guest OS are driving the VM to high CPU utilizations, there may be extended delays from the time the guest OS receives notification that receive packets are available until those packets are retrieved from the virtual NIC. In some cases, the high CPU utilization may in fact be caused by high network traffic, since the processing of network packets can place a significant demand on CPU resources.

- **Guest OS driver configuration.** Device-drivers for networking devices often have parameters that are tunable from within the guest OS. These may control behavior such as the use of interrupts versus polling, the number of packets retrieved on each interrupt, and the number of packets a device should buffer before interrupting the OS. Improper configuration of these parameters can cause poor network performance and dropped packets in the networking infrastructure.

Solutions

The solutions to the problem of dropped received packets are all related to ways of improving the ability of the guest OS to quickly retrieve packets from the virtual NIC. They are:

- **Increase the CPU resources provided to the VM.** When the VM is dropping receive packets dues to high CPU utilization, it may be necessary to add additional vCPUs in order to provide sufficient CPU resources. If

the high CPU utilization is due to network processing, refer to the OS documentation to ensure that the guest OS can take advantage of multiple CPUs when processing network traffic.

- **Increase the efficiency with which the VM uses CPU resources.** Applications which have high CPU utilization can often be tuned to improve their use of CPU resources. See the discussion about increasing VM efficiency in the Solutions section under Host CPU Saturation.

- **Tune network stack in the Guest OS.** It may be possible to tune the networking stack within the guest OS to improve the speed and efficiency with which it handles network packets. Refer to the documentation for the OS.

- **Add additional virtual NICs to the VM and spread network load across them.** In some guest OSes, all of the interrupts for each NIC are directed to a single processor core. As a result, the single processor can become a bottleneck, leading to dropped receive packets. Adding additional virtual NICs to these VMs will allow the processing of network interrupts to be spread across multiple processor cores.

Dropped Transmit Packets

Causes

When a VM transmits packets on a virtual NIC, those packets are buffered in the associated vSwitch port until being transmitted on the physical uplink devices. If the traffic from the VM, or from the set of VMs sharing the vSwitch, exceeds the physical capabilities of the uplink NICs or the networking infrastructure, then the vSwitch buffers can fill. In this case additional transmit packets arriving from the VM will be dropped. The count of these dropped packets over the selected measurement interval is available in the performance metrics visible from the vSphere client.

Solutions

In order to prevent transmit packets from being dropped, it is necessary to take one of the following steps:

- **Add additional uplink capacity to the vSwitch.** Adding additional physical uplink NICs to the vSwitch may alleviate the conditions that are causing transmit packets to be dropped. However, traffic should be monitored to ensure that the NIC teaming policies selected for the vSwitch lead to proper distribution of load over the available uplinks.

- **Move some VMs with high network demand to a different vSwitch.** If the vSwitch uplinks are overloaded, moving some VMs to additional vSwitches can help to rebalance the load.

- **Enhance the networking infrastructure.** In some cases the bottleneck may be in the networking infrastructure (e.g. the network switches or inter-switch links). It may be necessary to add additional capacity in the network to handle the load.

- **Reduce network traffic.** Reducing the network traffic generated by a VM can help to alleviate bottlenecks within the networking infrastructure. The implementation of this solution will depend on the application and guest OS being used. Techniques such as the use of caches for network data or tuning the network stack to use larger packets (for example, jumbo frames) may reduce the load on the network.

VMware Tools-Related Performance Problems

Overview

In order to achieve peak performance in a VMware vSphere environment, all VMs should have VMware Tools installed, up-to-date, and running properly. The proper procedure for installing VMware Tools in a guest OS is given

in the vSphere Guest Operating System Installation Guide. Installing VMware Tools will improve display, storage, and networking performance, as well as ensure that the vSphere's memory-management mechanisms operate to maintain proper performance on all VMs. Refer to the VMware Knowledgebase (http://kb.vmware.com/) if errors occur when installing VMware Tools.

The remainder of this section discusses the causes and solutions for specific VMware Tools-related performance problems. The existence of these observable problems can be determined by using the associated problem-check diagrams in the troubleshooting sections of this document. The information in this section can be used to help determine the root-cause of the problem, and to identify the steps needed to remedy the identified cause.

VMware Tools Not Running

Causes

Once VMware Tools has been installed in a VM, it will begin running whenever the VM is powered on and the guest OS has booted. However, there are a few situations which can prevent VMware Tools from starting properly after it has been installed. These are:

- **Changed or updated the OS kernel in a Linux VM.** In some cases, changing the kernel in a Linux VM, or updating the kernel to a more recent version, can lead to conditions that prevent the drivers that comprise the VMware tools from loading properly.

- **VMware Tools is disabled in the guest OS.** VMware Tools is installed as a set of services in a Windows VM, or as a set of device-drivers in a Linux VM. It is possible for those services/drivers to be deliberately disabled, or to be unintentionally disabled when performing other OS-level tasks.

Solutions

The solution to this problem will depend on the cause.

- **Changed or updated the OS kernel in a Linux VM.** In order to re-enable VMware Tools, rerun the vmware-config-tools.pl script which configures tools in a Linux VM. If this does not re-enable the tools, reinstall the tools.

- **VMware Tools disabled in the guest OS.** Re-enable the tools.

VMware Tools Out-Of-Date

Causes

As new versions of VMware ESX are released, VMware Tools is enhanced to take advantage of new features, or to improve the performance of existing features. If a VM is migrated to an ESX host running a more recent version of ESX, or if the version of ESX on the current host is updated, then the vSphere Client will report that the tools are Out-Of-Date in that VM.

Solutions

Update VMware Tools following the dialogs available in the vSphere Client.

Advanced Problems

High Guest Timer-Interrupt Rate

Causes

Most operating systems track the passage of time by configuring the underlying hardware to provide periodic interrupts. The rate at which those interrupts are configured to arrive varies for different OSes. High timer-interrupt rates can incur overhead which affects a VM's performance. The amount of overhead increases with the number of vCPUs assigned to a VM.

The cause of a high timer-interrupt rate may be one of the following:

- For many Linux OSes, the default timer interrupt-rate is high, ranging from 1000Hz to 5000Hz.
- For Microsoft Windows, some Java applications using the JVM from Sun Microsystems will raise the timer-interrupt rate from the default of 64Hz to 1000Hz.

For a general discussion of timekeeping in a VM and information on adjusting the timer-interrupt rate within common guest OSes, see the VMware technical paper *Timekeeping in VMware Virtual Machines* (http://www.vmware.com/resources/techresources/1066).

Solutions

Different mechanisms exist to reduce the timer-interrupt rate depending on the guest OS being used.

- For VMs running Linux, see the VMware knowledgebase article *Timekeeping best practices for Linux* (http://kb.vmware.com/kb/1006427).
- For VMs running Java applications on Microsoft Windows, see the VMware technical paper *Java in Virtual Machines on VMware ESX: Best Practices* (http://www.vmware.com/resources/techresources/1087) for a discussion of changing the way that Java affects the timer-interrupt rate.

Poor NUMA Locality

Causes

In a NUMA (Non-Uniform Memory Access) server, the delay incurred when accessing memory varies for different memory locations. Each processor chip on a NUMA server has local memory directly connected by one or more local memory controllers. For processes running on that chip, this memory can be accessed more rapidly than memory connected to other, remote, processor chips. When a high-percentage of a VM's memory is located on remote processor, we say that the VM has poor NUMA locality. When this happens, the VM's performance may be less than if its memory was all local.

All AMD Opteron processors and some recent Intel Xeon processors have a NUMA architecture.

When running on a NUMA server, the VMware ESX scheduler assigns each VM to a home node, which represents a single processor chip. In selecting a home node for a VM, the scheduler attempts to keep both the VM and its memory located on the same processor, thus maintaining good NUMA locality. However, there are some instances where a VM will have poor NUMA locality despite the efforts of the scheduler:

- **VM with more vCPUs than available per processor-chip.** The number of CPU cores in a home node is equivalent to the number of cores in each physical processor chip. When a VM has more vCPUs than there are logical threads in a home node, the ESX scheduler does not attempt to use NUMA optimizations for that VM. This means that the VM's memory is not migrated to be local to the processors on which its vCPUs are running, leading to poor NUMA locality.

- **VM Memory size is greater than memory per NUMA node.** Each physical processor in a NUMA server is usually configured with an equal amount of local memory. For example, a four-socket NUMA server with a total of 64GB of memory will typically have 16GB locally on each processor. If a VM has a memory size greater than the amount of memory local to each processor, the ESX scheduler does not attempt to use NUMA optimizations for that VM. This means that the VM's memory is not migrated to be local to the processors on which its vCPUs are running, leading to poor NUMA locality.

- **The ESX host is experiencing moderate to high CPU loads.** When multiple VMs are running on an ESX host, some of the VMs will end up sharing the same home node. When a VM is ready to run, the scheduler will attempt to schedule the VM on its home node. However, if the load on the home node is high, and other home nodes are less heavily loaded, the scheduler may move the VM to a different home node. Assuming that most of the VMs memory was located on the original home node, this will cause the VM to have poor NUMA locality. However, in return for the poor locality, the VM will experience an increase in available CPU resources. Moreover, the scheduler will eventually migrate the VMs memory to the new home node, thus restoring NUMA locality. As a result, temporary poor NUMA locality due to rebalancing by the scheduler is seldom, by itself, a source of performance problems. However, frequent rebalancing may be a symptom of other problems, such as Host CPU Saturation.

Solutions

NUMA locality can be improved by making changes to the configuration of the server or VMs. However, when making changes to improve NUMA locality, performance should be closely monitored to ensure that the changes lead to improvement. The impact of poor NUMA locality on performance will vary depending on the characteristics of the workload, and improving it will rarely yield performance gains larger than 10%. On the other hand, some of the solutions to poor NUMA locality may actually lead to worse performance.

The following steps can be taken to improve NUMA locality.

- **Reduce the number of vCPUs assigned to the VM.** If a VM has more vCPUs than there are cores on a processor chip, removing vCPUs to allow the VM to fit on a single package can improve NUMA locality. This has the obvious disadvantage of reducing the amount of CPU resources available to the VM. However, there are some situations in which this might improve performance:

    - If the VM was not using all available vCPUs, then removing some to improve NUMA locality may improve performance.

    - If the work being performed in the VM can be spread across multiple VMs with the same number of total vCPUs, but each fitting in a single NUMA node, then performance may be improved.

- **Increase the amount of physical memory on the host.** Adding additional physical memory on the host can allow VMs with large memory sizes, or multiple small-memory VMs, fit in the local memory of a single home node.

- **Reduce the size of VM memory to fit in a single NUMA node.** Reducing the amount of memory assigned to a VM so that it fits in the local memory of a single NUMA home node can improve NUMA locality and performance. However, reducing the amount of memory available to applications running in a VM may decrease their performance. The proper trade-off in this case will depend on the specific application, and can only be determined through testing.

- **In rare instances, enable memory-interleaving mode in BIOS.** In normal operation, the local memory for each processor chip in a NUMA system is assigned a contiguous range of memory addresses. This enables the scheduler to determine which memory is associated with each NUMA home-node, and thus to maintain

NUMA locality. In situations where many VMs, perhaps due to vCPU count or memory size, do not fit in a single home-node, overall performance may be improved by turning on memory-interleaving in the system BIOS.

With memory interleaving, the address for each consecutive line of memory is assigned to a different processor. For VMs whose memory footprint is too large to fit in a single node's memory, this ensures that at least some of the memory accesses are to local memory. Most NUMA systems have a BIOS setting to control whether they use interleaved or NUMA mode, although the exact terminology used may vary. Using memory interleaving may help the performance of a few large VMs at the expense of smaller VMs that fit in a single home-node. Switching to memory interleaving should be considered on as a last resort, and its impact on overall performance should be carefully tested.

For more information on the use of NUMA systems with ESX, and the NUMA scheduler, see the *vSphere Resource Management Guide*.

Performance Tuning for VMware ESX

Overview

This section will discuss some basic tuning parameters, either in the applications, guest OSes, or VMware ESX, that may be used to improve efficiency. They are discussed here to avoid repeating them in multiple places.

Using Large Memory Pages

Using large pages within an application and guest OS can improve performance and reduce virtualization overhead. See the VMware technical paper on Large Page Performance (http://www.vmware.com/resources/techresources/1039) for more information and instructions on enabling large pages.

Document Information

References

- vSphere Basic System Administration
  http://www.vmware.com/pdf/vsphere4/r40/vsp_40_admin_guide.pdf
- vSphere Resource Management Guide
  http://www.vmware.com/pdf/vsphere4/r40/vsp_40_resource_mgmt.pdf
- Large Page Performance
  http://www.vmware.com/resources/techresources/1039
- Timekeeping in VMware Virtual Machines
  http://www.vmware.com/resources/techresources/1066
- Timekeeping best practices for Linux
  http://kb.vmware.com/kb/1006427
- Java in Virtual Machines on VMware ESX: Best Practices
  http://www.vmware.com/resources/techresources/1087

Change Information

- Current Version: V1.0. Original version.
- Previous Versions: None

About the Author

Hal Rosenberg is a performance engineer at VMware. His focus areas are Java, VDI/terminal services, and performance troubleshooting. Prior to coming to VMware, he has over 10 years of experience working on performance engineering and analysis for hardware and software projects at IBM and Sun Microsystems.

VMware, Inc. 3401 Hillview Ave., Palo Alto, CA 94304 www.vmware.com