



Eliminating VMware / Storage Related Performance Challenges with VirtualWisdom®

Contents

Introduction 3

The Beginning of the Performance Problem Cycle.....5

Looking Beyond Device-Specific and Averaged Out Metrics 6

The Need for a Comprehensive View that Incorporates Exchange Completion Times..... 7

Historical Trending and Playback That Goes Beyond Disk I/O Metrics 8

A More Accurate Way to Determine VM to ESX ratios..... 9

De-risking Automated vMotions via DRS 10

Ensuring the I/O for SIOC 11

Frames/Sec and ECTs as Opposed to Just MB/s and IOPS..... 11

Ensuring That Multipathing is Reducing, Not Adding Risk 12

Correlating Issues from the VM to the LUN 21

Conclusion 22

Introduction

Whether on a virtualized or non-virtualized platform, application performance is heavily affected by its underlying storage infrastructure. The complexity of correctly configuring storage in accordance to application demands can range from deciding the right RAID level, number of disks per LUN, array cache sizes to the correct queue depth and fan-in / fan-out ratio. These and other variables can drastically influence how I/O performance is handled and ultimately how applications respond. With virtualized environments, the situation is no different, with storage related problems often being the cause of most VMware infrastructure mis-configurations that inadvertently affect performance.

This document addresses specifically how to proactively deal with the performance and troubleshooting related issues found in a Fibre Channel SAN / vSphere environment. Using Virtual Instruments' revolutionary platform, VirtualWisdom, this document showcases the need for a comprehensive and real-time view to proactively deal with performance issues in a virtual server and SAN environment. By initially identifying the common storage/VMware related performance problems, we will demonstrate how, by using VirtualWisdom, these problems could have been quickly identified and often eliminated before they occurred.

This document also proposes a new way of proactive performance monitoring and optimization via VirtualWisdom that reduces troubleshooting from days to minutes. For the performance management requirements of Tier 1 applications, we discuss how the unique granular level of I/O performance provided by VirtualWisdom is an essential element for performance optimization.

Additionally, we focus on some of the new vSphere initiatives which were developed to enhance the storage / vSphere stack, such as VAAI and SIOC. We demonstrate how VirtualWisdom meets the need for a comprehensive view of the environment (i.e. from Initiator to Target to LUN). We explain how visibility into the SAN fabric enables the optimization of storage systems servicing virtual machines.



VirtualWisdom Architecture & Deployment

To fully understand the content and conclusions herein, it is important to first understand Virtual Instruments' VirtualWisdom. VirtualWisdom consists of the following software and hardware components:

The VirtualWisdom SAN Availability Probe collects status from SAN switches via SNMP. It collects and records metrics for

each port. These metrics are compiled and analyzed, and made available through the VirtualWisdom dashboard. Metrics include the number of frames and bytes, as well as the key faults for each port e.g. loss of synchronization, link resets, link failures, Class 3 discards and CRC errors.

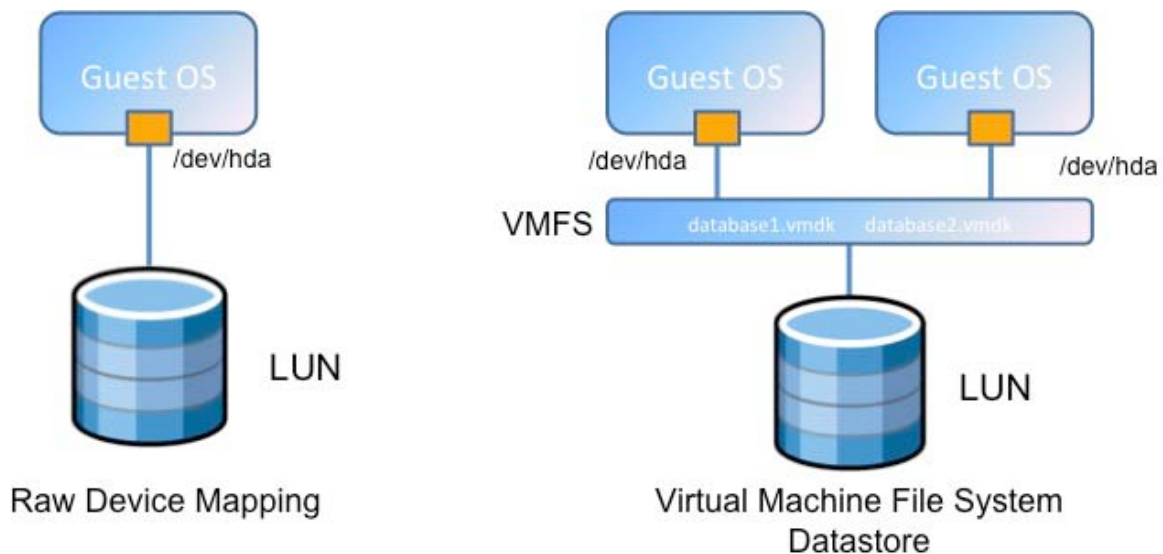
The Virtual Server Probe is a software probe that collects status from VMware's ESX servers via vCenter. It collects data on and calculates 120+ different metrics including CPU utilization and status, memory utilization, disk I/O requests and capacity, network requests and utilization.

The VirtualWisdom SAN Performance Probe is a Fibre-Channel hardware probe that analyzes every frame header on a fibre-channel SAN link. The SAN Performance Probe detects application performance slowdowns and transmission errors by measuring every SCSI I/O transaction from start to finish, for every server/volume combination (initiator/ target/ LUN).

The Traffic Access Points (TAPs) provide a passive, non-intrusive access point to the fibre-channel network traffic on the SAN for failure analysis, problem diagnosis and performance monitoring. TAPs operate "out-of-band" by transparently diverting some of the optical signal through the TAP to another port, which provides a copy of the Fibre Channel frame headers. The agent-less TAP has no impact on application or SAN performance and is integrated with a Fibre Channel Patch Panel for simple deployment.

RDM or VMFS Configuration of FC SAN storage?

It is important to recap on the basics of how storage is provisioned in a FC SAN VMware environment:



The most commonly used option for vSphere storage configuration is what is termed the Virtual Machine File Store or VMFS. In this method, several Virtual Machines are able to access the same LUN or a pool of LUNs. The immediate advantages of such an approach are first in terms of provisioning and zoning. This becomes far more simplistic as opposed to a one-to-one mapping ratio that is required for each LUN for each Virtual Machine. Additionally, this makes backups far easier as the VMFS for the given Virtual Machines need only be dealt with instead of numerous individual LUNs that are mapped to many Virtual Machines. VMFS volumes can be as big as 2TB and with the concatenation of additional partitions which are termed VMFS extents, this can then be as large as 64TB, i.e. 32 extents. With extents, it's usually best practice to create these on new physical LUNs to manage additional LUN queues or throughput congestion. Coupled with this, extents should be assigned the same RAID and disk type to avoid performance pitfalls. In general, datastore sizing is still a rule of thumb exercise with compromising tradeoffs between deciding whether to go with few large 2TB datastores or many small datastores. Ultimately though, despite its popularity, scalability and ease of use, VMFS is the option that has the potential to cause the most performance problems due to mis-configuration.

The VM administrator also has the option of Raw Device Mapping (RDM). One of RDM's key features is

that it allows the use of native storage arrays' capabilities. On the downside, RDM does somewhat restrict the use of vMotion.

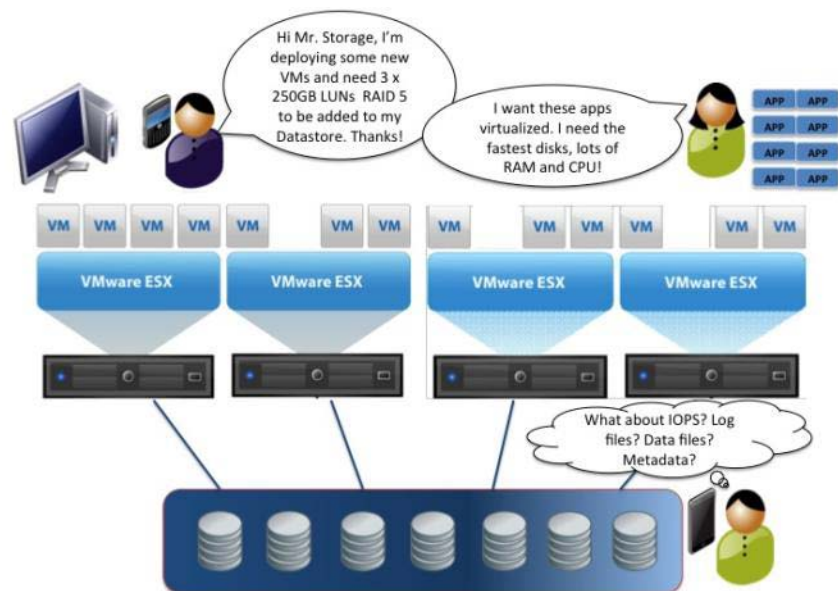
There are two types of RDM, physical and virtual, and in its simplest sense RDM is merely a direct VM to LUN relationship that is provisioned in a manner similar to how a storage administrator would map to a physical server. All of the metadata required to manage and proxy the disk access is contained within the RDM file, ordering the VMkernel where to send disk instructions. The main distinction between physical and virtual RDMs is that only the latter allows for the use of snapshots.

RDMs are chosen for a number of reasons and one of their main advantages is that they greatly aid troubleshooting and enable the storage administrator to provision and align the right types of LUNs to the Virtual Machine's specific application requirements. For a transactional database that heavily relies on its disk performance, Raw Device Mapping provides a fairly simple method to provision separate LUNs with different RAID levels and IOP requirements for the database's data and log files. Furthermore, if admins wish to deploy NPIV to ensure a more granular insight into the performance metrics of a given VM, RDM is the only supported method. (N_Port ID Virtualization or NPIV is a Fibre Channel facility allowing multiple N_Port IDs to share a single physical N_Port. This allows multiple Fibre Channel initiators to occupy a single physical port, easing hardware requirements in Storage Area Network design, especially where virtual SANs are called for. To fully use NPIV, you must have NPIV capable HBAs and switches). Despite this advantage, due to their capacity planning and scalability restraints (i.e. a separate LUN is required to be provisioned for each new RDM), seasoned VM administrators use this method sparingly.

The Beginning of the Performance Problem Cycle

The VMFS / poorly configured storage conundrum has led to arguably one of the major challenges to server virtualization, what is termed as the 'Virtual Stall'. For various reasons there is a reluctance to deploy Tier 1 / Mission Critical applications on vSphere. Such applications are often linked with predetermined formal or informal SLAs. It is when these SLAs are not met that a performance problem is flagged, often with vSphere being falsely blamed. Once performance slows, the troubleshooting and finger-pointing process commences. We should note that the majority of storage related performance problems are in fact initiated at the beginning of the provisioning process or even earlier, at the design phase.

The cycle typically starts when application owners requesting Virtual Machines begin to ask for over-provisioned resources to ensure they have no performance issues. This in turn exerts pressure on VM administrators to overprovision Memory and CPU to avoid any potential application slowdowns being falsely ascribed to vSphere. In terms of storage, the VM administrator will falsely think along the lines of capacity that needs to be added to their VMFS and subsequently request the amount of storage they feel is best to satisfy the performance requirements of their Virtual Machines. At best, they may request the RAID level and the type of storage, e.g. 15K RPM FC disks. It is here that the discrepancy arises for the storage administrator. The



storage admin, used to provisioning LUNs on the basis of application requirements, will instead not be thinking of capacity but rather in terms IOPS and RAID levels. Eventually, though as there is no one-to-one mapping and the requested LUN is to be merely added to a VMFS, the storage administrator, not wishing to be the bottleneck of the process, will proceed to add the requested LUN to the pool. Herein lies the source of a lot of eventual performance problems, as overtly busy LUNs begin to affect all of their aligned virtual machines as well as those that share the same datastore. Moreover, if the LUN is part of a very busy RAID group on the backend of the storage system, such saturated I/O will impact all of the related physical spindles and all of the LUNs they share. What needs to be appreciated is that the workload of individual applications presented to individual volumes will be significantly different to that of multiple applications being consolidated onto a single VMFS volume. The numerous I/Os of multiple applications alone, even if sequential, will push the storage system to deal with these numerous requests as random, thus requiring different considerations than those for individual applications, such as RAID level, LUN layout, cache capacity, etc. Of course, good provisioning practices can obviate this problem, but in a rush to respond to server-based requests, it's not unusual to follow the easiest and fastest path.

Due to this, the storage I/O stack of the vSphere architecture now takes a lot more precedence compared to the days of ESX 2.x when there was no GUI visibility for storage bottlenecks. With visibility into disk I/O being introduced into ESX 3.x, VM administrators were at last able to investigate poorly provisioned LUNs that were so busy that their I/O bottlenecks were affecting the performance of the datastore in which they resided. Indeed, the storage I/O bottlenecks would be so detrimental to Virtual Machine performance, that VM administrators would be engaged in days if not weeks of troubleshooting and resolving such issues. So how do such issues arise, how could they be avoided and what are the solutions that remediate such performance problems in a quick and proactive manner? How do you bring true visibility and insight into vSphere's storage I/O stack?

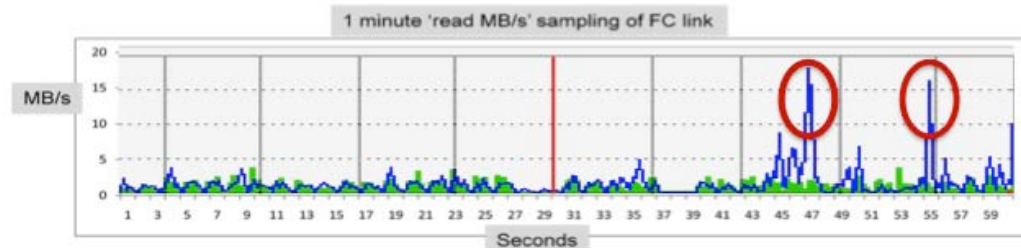
Looking Beyond Device-Specific and Averaged Out Metrics

To counter such problems there is a customary troubleshooting procedure which VM and storage administrators often follow. This procedure uses metrics found in vCenter: esxtop, vscsiStats, IOMeter, Solaris IOSTAT, PerfMON and the array management tool. This somewhat laborious process usually includes measuring the effective bandwidth and resource consumption between the VM and storage, then using other paths between the VMs and storage, and even reconfiguring cache and RAID levels. This often takes days if not weeks spent in checking for excessive LUN and RAID group demands, understanding the VMFS LUN layout on the backend of the storage's physical spindles, and investigating the array's front end, cache and processor utilization as well as bottlenecks on the ESX host ports. Some admins may also attempt to optimize Queue Depth settings, which without an accurate measurements, is at best a guessing game based on rule-of-thumb.

Despite all of these measures, there is still no guarantee that this will identify or eliminate performance issues, leaving vSphere to be erroneously blamed as the cause. Or the application is unfairly deemed 'unfit' to be virtualized. The underlying reason for this is that the aforementioned metrics are still only device-specific and averaged out samples. Therefore, a true and comprehensive viewpoint from VM to LUN in real time I/O is missing from any of these troubleshooting scenarios. This often leads to a large element of guesswork, and potentially incorrect conclusions.

Currently, despite the plethora of VM/storage related products and tools in the market, the performance metrics they provide on VM memory, CPU, Network and Disk I/O are all based on polling and dependent on metrics averaged over the sampling period, usually from vCenter, the storage system, or both. The conundrum with such device-specific views is that true I/O latency is often missed, especially when these issues

occur

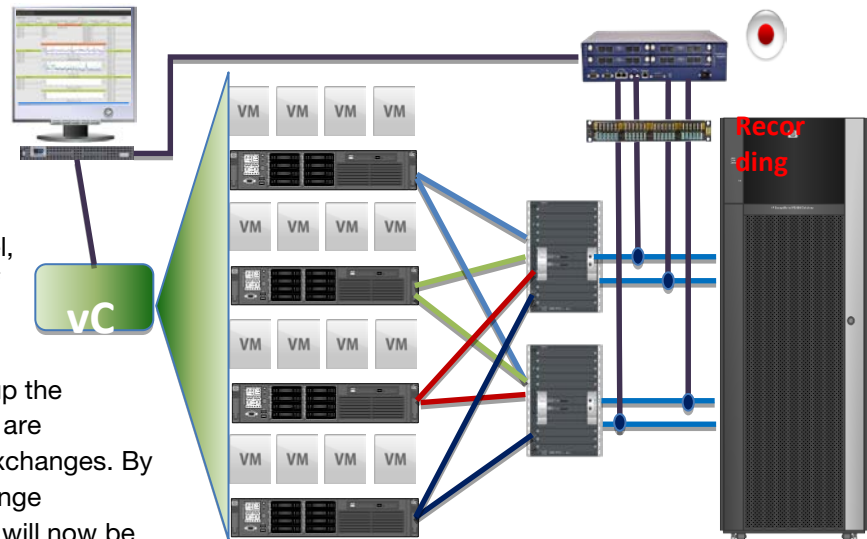


intermittently or only during peak periods, and therefore beyond the scope of the ‘averages’. At best, such tools may give you the highest peak in that averaged out time frame, but even this misses the true picture of what is occurring. For example, looking at the graph below of a 1 minute interval, this would characteristically be reported by either vCenter or a storage management tool as an average of 5MB/s. The highlighted brief peaks of 20MB/s would be missed. With such incidents usually being intermittent, unless a more accurate reading takes place, such problems will fail to be diagnosed and lead to troubleshooting nightmares between the VM and storage administrators.

Indeed, it is these same averaged metrics that are the basis of VM capacity planning tools and so often there is still a rule-of-thumb aspect involved in the determination of how many VMs to deploy on a given ESX server. With “VMware first” policies being adopted by the mainstream, the challenge is not whether to virtualize but how much can be virtualized without sacrificing performance. Furthermore, there’s an ever-increasing recognition of Tier 1 applications which need to be virtualized. Ensuring that the physical to virtual migration process doesn’t bring with it performance problems is an imperative. And it simply can’t be guaranteed with the reliance on averaged metrics that fail to give a comprehensive view of the I/O. So to achieve this, a comprehensive end-to-end real-time view of the storage and virtual infrastructure which also incorporates the SAN fabric is required.

The Need for a Comprehensive View that Incorporates Exchange Completion Times

To fully understand application requirements prior to virtualization and have an insight into storage related performance problems, it’s vital to have a solution that reports in real time and on the Exchange Completion Time. In fibre channel, the FC Frame is the equivalent of what IP terms a Packet. Several of these Frames make up a Sequence, which in turn makes up the Exchange. SCSI I/O transactions are encapsulated in Fibre Channel Exchanges. By being able to measure the Exchange Completion Time, administrators will now be empowered with the knowledge of the exact amount of time it takes for a complete Exchange, i.e. the time that all frames and sequences have passed from the host to the storage port to the LUN and acknowledged back to the host. Coupled with the metrics taken from vCenter via the Virtual Server Probe, from the switches via the SAN Availability Probe, and from the FC frames via the SAN Performance Probe, VirtualWisdom provides a comprehensive, real-time monitoring system that is out of band and able to report on application I/O latency down to the sub-millisecond level. This ability to accurately measure response time also bridges the gap

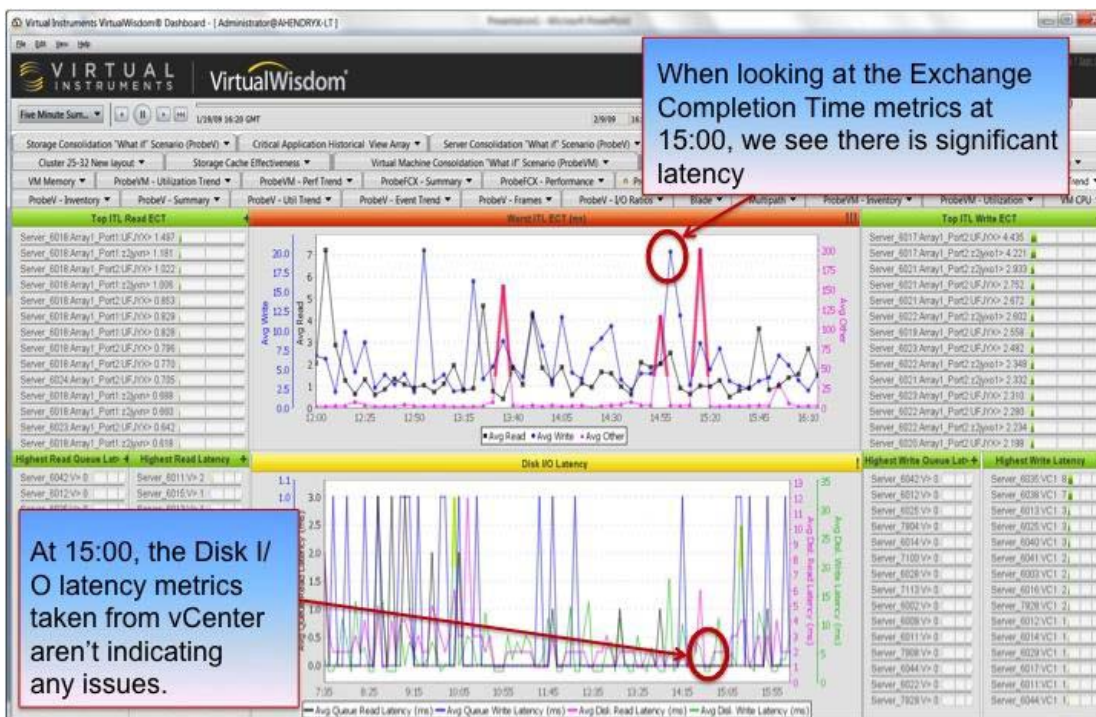


between the application, server and storage silos. Previously, an application owner would request storage based on capacity, and the storage engineer would link this to IOPS and MB/s. Instead, an application owner can request a LUN's response time, which the storage engineer is now able to determine by observing the Exchange Completion Times.

With every single FC frame header being analyzed, the VM and storage administrator, together with the application owner, can work together to optimize virtualize applications. With VirtualWisdom, they not only have a common language of response times and a comprehensive view of their applications but also an accurate knowledge of their I/O latency. This is significantly different than the current tools being administered by application owners that will merely report that an application is suffering from latency. Because these tools are host specific, they fail to tell you where the latency is actually occurring because beyond the HBA, they have no insight. By knowing the Exchange Completion Times and analyzing each FC Frame Header, latency can not only be reported, but the root cause can be pinpointed, whether it's an extremely busy storage port, a flapping HBA or a bent cable in the SAN fabric. Additionally you can use the VirtualWisdom's historical playback feature to know how an application performs during different periods. This can be used as a baseline to measure the application's requirements and how it performs once it's been virtualized.

Historical Trending and Playback That Goes Beyond Disk I/O Metrics

With the knowledge of Exchange Completion Times, long, laborious and often aimless troubleshooting can be avoided. For example, in the graph below, VirtualWisdom reports the disk I/O from vCenter having no issues at 3pm. When we look at the corresponding and more accurate Exchange Completion Times, we are in fact seeing significant latency.

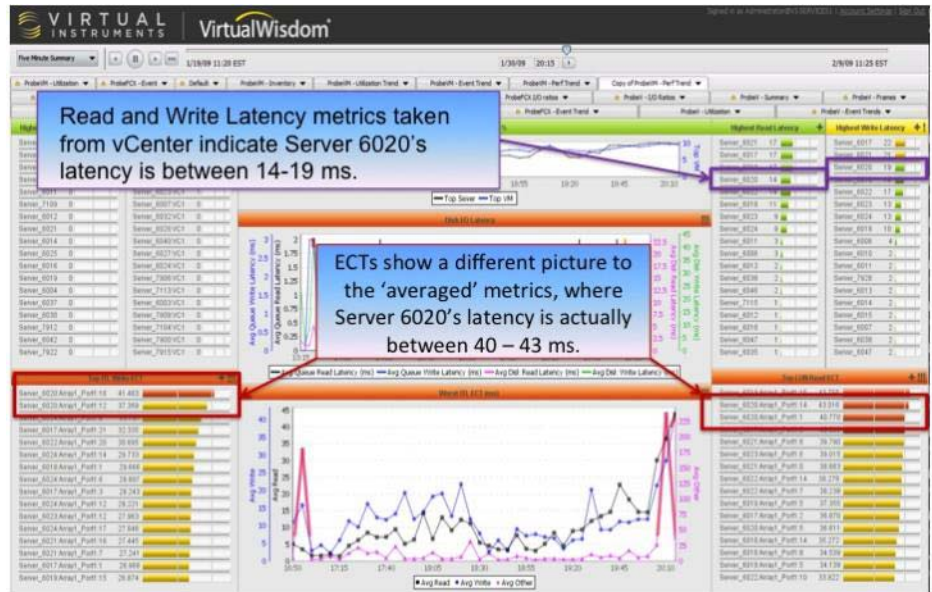


By using the timeline playback feature and rewinding back to 3pm, we are able to see exactly what happened with regards to where the latency is occurring, i.e. between which initiator, target and LUN. This is something that disk I/O alone would fail to identify, causing major troubleshooting and performance headaches.

For example, in the screenshot below we have the read and write metrics from vCenter indicating latency of the given ESX servers to be 14-19ms. The more accurate Exchange Completion Time gives a

radically different reading of 40-43ms. With a Tier 1 application that's been virtualized, such metrics are vital in eliminating potential slowdowns or worse still, unplanned downtime.

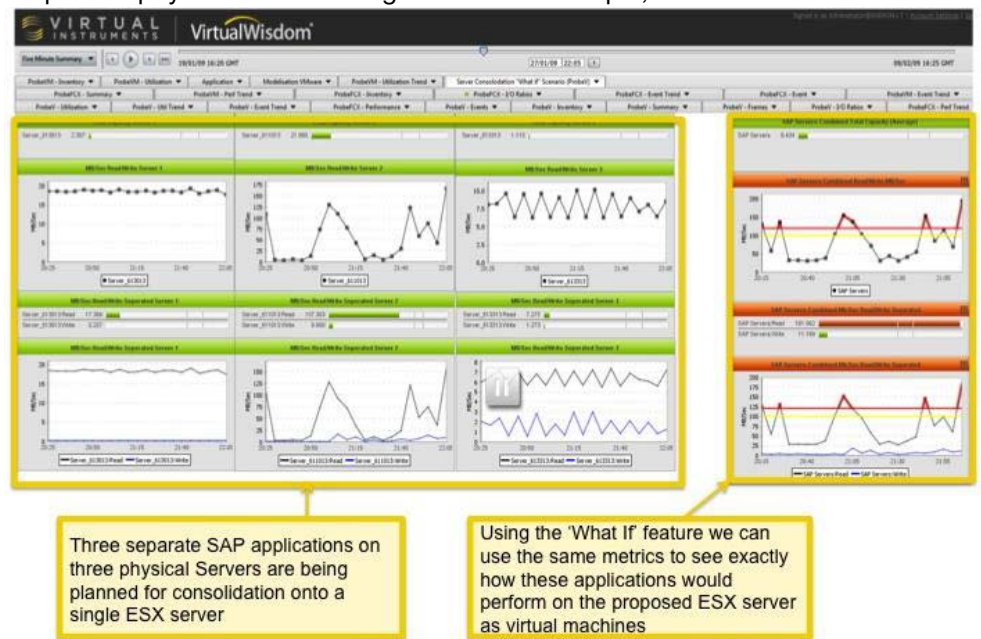
In fact, such discrepancies are common in situations where host-based measurements are relied upon. Additionally, if monitoring is based on a host based tool, such a tool would use server CPU and memory, which in itself causes latency. Server administrators are often falsely led to believe there's latency occurring elsewhere when in fact it's the servers themselves that have become overloaded. An end-to-end comprehensive view is the solution to this problem.



A More Accurate Way to Determine VM to ESX ratios

The insight provided by such a comprehensive view also enables the ability to accurately determine the correct ESX to VM ratios prior to physical to virtual migrations. For example, VirtualWisdom offers the ability to run 'what if' simulations

which use real historical metrics and data to allow the admin to model and see the effects of potential changes. For example, as seen in this screenshot, a modeling dashboard can be set up as an alternative to setting up a new lab or test environment to see

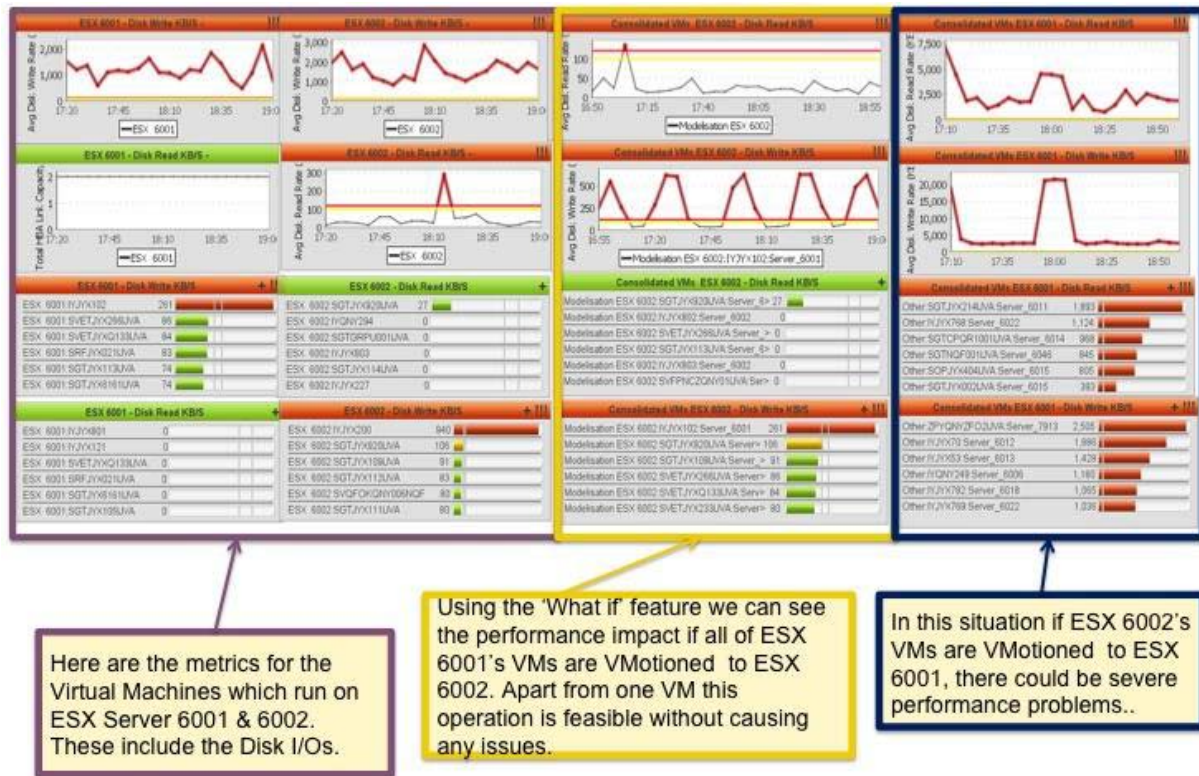


how an application would perform on a virtual platform. Here we have chosen particular metrics of three separate physical servers that are running three separate SAP applications. Prior to migrating them, using the actual metrics of the applications, we have decided to run a modeling configuration which shows us exactly how those applications would perform once virtualized onto a single ESX server. This can then be historically tracked back to see how the proposed ESX server would have performed at different peak periods for that application, eliminating risk and making the most of the server's

resources. By being able to draw on historical metrics such as Disk I/O, MB/s, CPU utilization etc., such a modeling example can significantly reduce the risk of deploying Tier 1 applications onto vSphere.

De-risking Automated vMotions via DRS

Using vMotion, VMware's Distributed Resource Scheduler (DRS) automatically moves Virtual Machines to other ESX servers if resources become scarce. DRS is an integral feature in maintaining business continuity in virtual set ups, but one of its limitations has always been that it doesn't take into consideration the storage stack. Couple this with a potential scenario of an admin not setting DRS to its optimum configuration, and you run the risk of VMs being automatically vMotioned in an attempt to avoid performance problems only to face more or different performance problems on a different ESX server. In a situation where a VM is critical, this is a precarious position to be in, and one that risks vSphere being falsely blamed for underlying issues such as poor storage configuration or lack of available resources. To alleviate such a situation, a modeling dashboard as the one shown below could be set up.



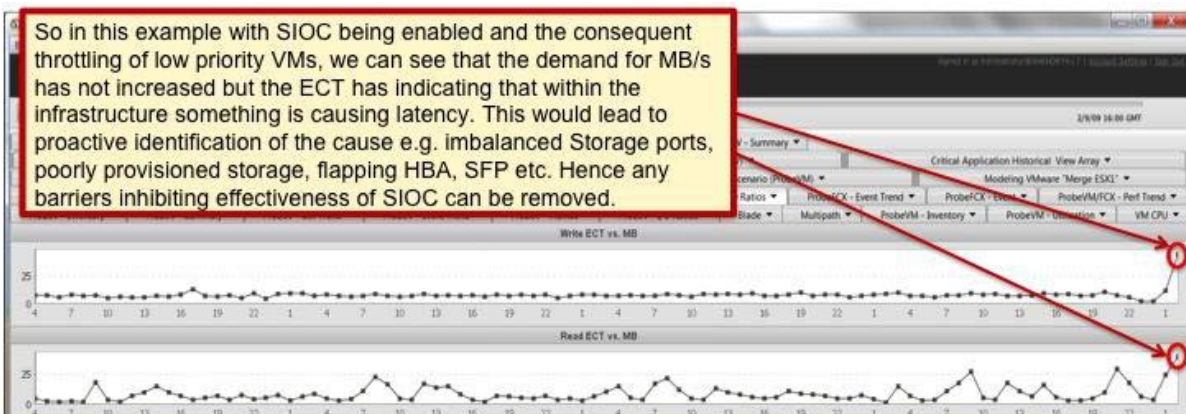
Here we have several virtual machines residing on ESX servers 6001 and 6002. In this example, the disk I/O metrics have been used, but of course other metrics could have also been used such as memory ballooning, network utilization etc. In this modeling example, we can foresee the effects of vMotioning all of ESX server 6002's VMs to ESX server 6001. In the second example we can foresee the effects of how ESX server 6002 would perform if all of the VMs from ESX 6001 are vMotioned.

A potential use of such a modeling dashboard is where the admin has set a DRS threshold of 80% for a critical VM. They can then also concurrently set an alarm on VirtualWisdom to alert when the VM reaches a 60% threshold. Using this dashboard, we can quickly see the effect of vMotioning the critical VM before it's automatically done so by DRS. So if an automated vMotion will cause further problems, the DRS set up can be quickly rearranged. Furthermore, prior to adding any VMs to a DRS cluster and assigning share levels, one can now optimize resources by knowing beforehand how those VMs would behave within the cluster.

Ensuring the I/O for SIOC

With VMs having to compete for shared storage resources, the performance of their critical applications begin to suffer especially when several I/O-intensive applications issue a very high number of requests concurrently. To counter such situations, VMware have introduced what is termed storage I/O control (SIOC). The aim of SIOC is to identify disk I/O latency at the VMFS volume level and in a similar way to DRS, assign shares and priorities to protect the high priority VMs. By continuously monitoring the aggregate normalized I/O latency across the VMFS datastores, SIOC will detect the I/O congestion based on preset thresholds. So in a similar way to DRS, SIOC will then throttle a VM's throughput once a volume's normalized latency crosses that preset threshold. The net effect of this is that under contention, SIOC will ensure that VMs with higher shares will be given priority access to the storage. So while SIOC can bring great dividends in solving storage / vSphere related problems, if there are still underlying problems with the storage configuration or the SAN fabric, the benefits of SIOC will still not be fully realized.

For example, in the dashboard below we can see that while SIOC has been enabled, we see that the Exchange Completion Times are increasing. This indicates that the VM that has been given priority is

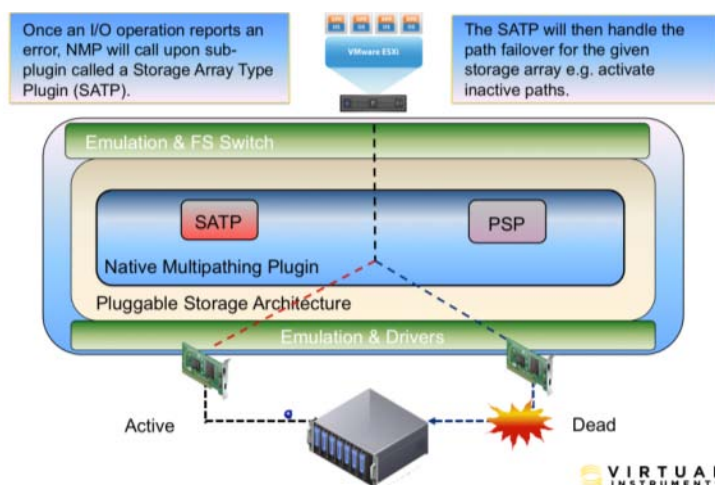


still suffering from latency. This could be caused by poorly configured storage, imbalanced storage ports, or issues with the SAN fabric. Such underlying issues would inhibit the benefits of SIOC and still cause major latency to the prioritized VM. Without the ability to measure, you won't know the real problem(s).

Frames/Sec and ECTs as Opposed to Just MB/s and IOPS

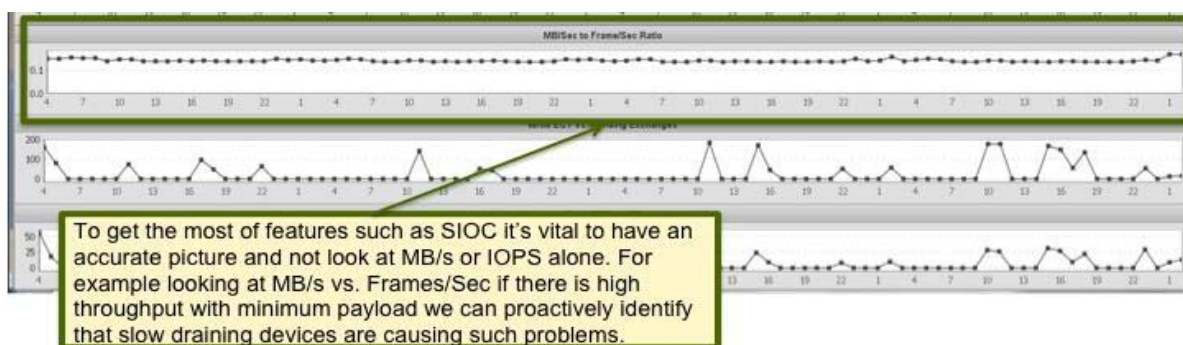
While many storage arrays have tools that provide information on IOPS and MB/sec that can be correlated with the metrics seen on the ESX level, if we look at a metric such as the ratio of frames/sec to MB/sec which is provided by VirtualWisdom, we will actually get a better picture of the environment and its underlying latency.

For example, if we are seeing high throughput but minimum payload, we can proactively identify whether there are a number of management frames (instead of data) which are dealing with issues such as logins and logouts, loss of sync or some other



optic degradation. And we can see this prior to causing major problems.

To elaborate, the MB/sec to Frames/Sec ratio that is seen below in this VirtualWisdom screenshot is different than the IOPS metric. Referring back to the FC Frame, a Standard FC Frame has a data payload of 2112 bytes, i.e. a 2K payload. So for example, an application that has an 8K I/O will require 4 FC Frames to carry that data portion. This would equate to 1 IOP being 4 Frames and subsequently 100 IOPS of the same size equating to 400 Frames. To get a true picture of utilization looking at IOPS alone is not sufficient because there exists a magnitude of difference between particular applications and their I/O size, with some ranging from 2K to even 256K. With backup applications, the I/O sizes can be even larger. So with reference to the graph below of MB/sec to Frame/sec ratio, the line graph should never be below the 0.2 of the y-axis i.e. the 2K data payload. If the ratio falls below this, for instance, at the 0.1 level, we can identify that data is not being passed efficiently despite the throughput being maintained (MB/sec). This is typical of when management frames are being passed instead of data, often reporting on link resets or loss of syncs that are occurring. With VirtualWisdom, you can customize an alarm be informed when the Frames are no longer carrying data. This allows you to proactively identify and



remediate the cause before major performance setbacks occur.

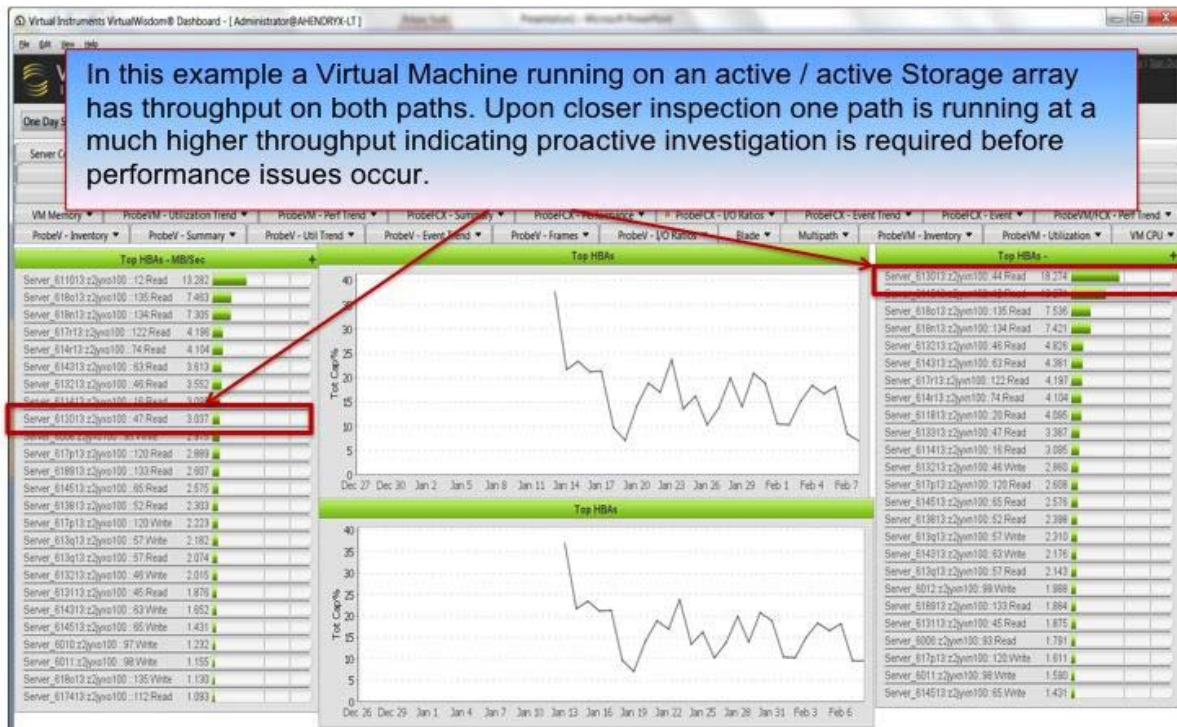
Exchange Completion Times should be the standard for measuring VM to storage performance, not MB/s and IOPS. To use an analogy of postmen who must deliver their packages, MB/s would be akin to measuring how many postmen are travelling with packages. This is not an accurate measure as many of those packages may be taking a long time if at all to be delivered. A far better measurement would be to know how long each postman takes to pick up his package, deliver it to its destination and return back for another. This is what would equate to an Exchange Completion Time and a true measure of latency and VM/storage performance.

Ensuring That Multipathing is Reducing, Not Adding Risk

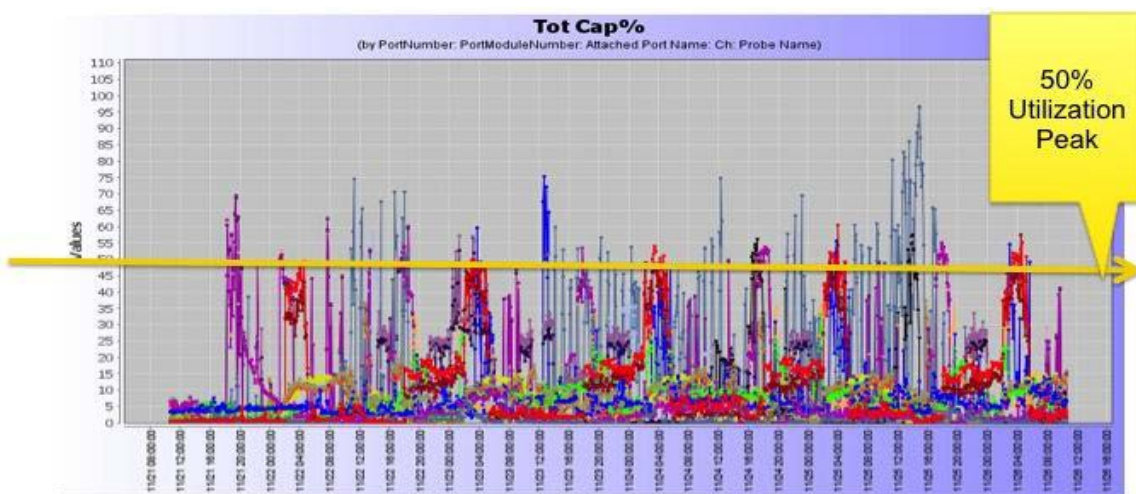
One of the challenges presented with vSphere and VMFS is that a single datastore can only have one active I/O path at any given time, thus not allowing bandwidth to be aggregated across several paths. Considering this limitation and that most Enterprise storage systems are active/active, hence simultaneously receiving I/O from multiple concurrent paths to their LUNs, VM admins have to choose the active path on a LUN-by-LUN basis. To achieve this, the multipathing policy that they would have to use would be 'Fixed'. With an active/passive storage system, only one storage processor can accept I/O for a particular LUN with the other storage processor being required for failover. With the ESX hosts able to discover the active paths with such storage systems, the best multipathing policy to use for this situation is Most Recently Used (MRU). It is often here where performance issues occur, as path policies are wrongly assigned in to their corresponding storage system. For example, if instead of MRU, a Fixed policy is mistakenly assigned to an active/passive storage system, the storage system would subsequently suffer from LUN path thrashing.

With recent vSphere versions, a new storage layer was introduced named the Pluggable Storage Architecture (PSA) that allowed hosts to use third party multipathing software. Furthermore, with the

absence of any third party APIs, hosts can utilize what is termed the NMP (Native Multipathing Policy). This has now made load balancing and multipathing a lot less problematic. For load balancing, the pluggable storage architecture (PSA), which is basically the VMkernel API, allows the 3rd party code to be inserted for the ESX I/O path. The PSA is also responsible for coordinating with the Native Multipathing Plug-in. So as can be seen in the diagram below, once a VM issues an I/O request to the storage, the



Native Multipathing Plug-in will call upon the Path Selection Plug-in (PSP). The PSP then chooses the path for the I/O request. For failover, i.e. in a situation when a path has gone down, the NMP will instead call upon a subplugin called the SATP or the Storage Array Type Plug-in. SATP will then deal with the failover and activate the inactive path allowing the I/O request to be dealt with. Additionally, the host associates with the SATP which kind of storage system it's dealing with, i.e. active/active or active/passive, and uses this information to automatically set the correct pathing policy for each LUN. Based on this, the PSP is also automatically chosen, helping to eliminate the incorrectly chosen path policies of the past.



Furthermore, in an active/active Round Robin set up it's vital to be aware that certain HBAs aren't spiking over 50%, to avoid the disaster of a failover that can't take the load, leading to a potential outage. Looking at the screenshot above we can see how VirtualWisdom facilitates this. With customized alarms, VirtualWisdom can proactively alert the admin when and where such incidents occur. To ensure you get the best from the Multipathing APIs, guaranteeing that your HBA throughput is balanced is an essential starting point.

Additionally, when storage or SAN ports are severely overloaded or imbalanced, serious application-level performance problems are a usual consequence, whether in a virtualized or non-virtualized environment. VirtualWisdom provides the ability to quickly detect or be alerted of such imbalances, ensuring that the I/O load can proactively be spread or balanced across multiple devices, satisfying the I/O demand. For example, when there are heavy IO loads, VirtualWisdom can proactively alert the admin when bottlenecks are about to be caused either by the storage processor or their corresponding storage ports. With this insight, such loads can quickly be distributed across multiple storage processors or storage ports without sacrificing the performance of other applications.

The Best Method of Setting Optimum Queue Depth

A storage system's Queue Depth in its most basic terms is the physical limit of exchanges that can be open on a storage port at any one time. The Queue Depth setting on the HBA will specify how many exchanges can be sent to a LUN at one time. Generally, most admins leave their Queue Depth settings at the manufacturer's default with only the requirement to facilitate a small number of I/O intensive VMs leading them to make an increase. The risk of changing or not changing Queue Depths to their optimum can have severe detrimental effects on performance, where any outstanding I/O queuing can cause bottlenecks.

For example, if Queue Depth settings are set too high, the storage ports will quickly become overrun or congested, leading to poor application and VM performance. If the storage port gets full, new SCSI commands will be responded to with a QFULL status. This will cause the VMkernel to throttle back I/O to the storage port in an aggressive fashion, dramatically reducing throughput and impacting VM performance. Outstanding commands can also back-up across the SAN leading to congestion on ISL links, which may impact SAN traffic. In extreme cases, this can cause data corruption or loss.

If Queue Depth settings are set too low, the storage ports become underutilized, leading to poor SAN efficiency. On the other hand, should the Queue Depth be correctly optimized, performance of VMs and their corresponding LUNs can be vastly improved..

Generally, VM Admins use 'esxstop' to check for I/O Queue Depths and latency with the QUED column showing the queuing levels. With VirtualWisdom, admins are now empowered with the only solution that can measure real-time aggregated queue depth regardless of storage vendor or device, i.e. in a comprehensive manner that takes into consideration the whole process from Initiator to Target to LUN. VirtualWisdom's unique ability to do this ensures that storage ports are optimized for maximum application health, performance, and SAN efficiency.

It is important to prevent the storage port from being over-run by considering both the number of servers that are connected to it as well as the number of LUNs it has available. By knowing the number of exchanges that are pending at any one time it is possible to manage the storage Queue Depths.

In order to properly manage the storage Queue Depths one must consider both the configuration settings at the host bus adapter (HBA) in a server and the physical limits on the storage arrays. It is important to determine what the Queue Depth limits are for each storage array. All of the HBAs that access a storage port must be configured with this limit in mind. Some HBA vendors allow setting HBA and LUN level Queue Depths, while some allow HBA level setting only.

The default value for the HBA can vary a great deal by manufacturer or by operating system and version, and are often set higher than what is optimal for most environments. If you set the Queue Depths too low on the HBA it could significantly impair the HBA's performance and lead to under-utilization of the capacity on the storage port (i.e. underutilizing storage resources). This occurs both because the network will be underutilized and the storage system will not be able to take advantage of its caching and serialization algorithms that greatly improve performance. Queue Depth settings on HBAs can also be used to throttle servers so that the most critical servers are allowed greater access to the necessary storage and network bandwidth.

To deal with this, the initial step should be to baseline the virtual environment to determine which servers already have their optimal settings and which ones are either set too high or too low. Below is a sample VirtualWisdom table showing real time Queue Depth utilization during a reporting period. Here we can see all of the initiators and the maximum queue depths that were recorded during the recording period. This table can be used as a method to compare the settings on the servers to the relative values of the applications that they support. The systems that are most critical should be set to higher Queue Depths than those that are less critical; however Queue Depth settings should still be within the vendor specified range. Unless storage ports have been dedicated to a server, VirtualWisdom often shows that optimum Queue Depth settings should be between the ranges of 2-8, despite industry defaults tending to be between 32-256. Note that with high performance or virtualized storage, these can often be set higher.

To explain this further, in the report below we have a column in descending order of the Maximum Pending Exchanges

and their corresponding initiators and server names. The Maximum Pending Exchanges are not only the maximum number of exchanges pending during the interval being recorded but also the exchanges that were opened in previous intervals that have not yet closed.

Initiator	Probe Name	Initiator Name	Max Pending Exchanges
0x300047	SVCNode2C2P1	ESXServer1	256
0x300047	SVCNode1C2P1	ESXServer2	256
0xea0031	SVCNode1C2P2	ESXServerUAT	254
0xea0002	SVCNode2C1P1	ESXServer3	254
0x30003f	SVCNode1C1P2	ESXServerDEV	254
...

So for example, if a report such as this was produced for 100 ESX servers, it's important to consider whether your top initiators are hosting your highest priority applications and whether your initiators with low Queue Depth settings are hosting your lowest priority applications. Once the appropriate Queue Depth settings have been determined, it is a best practice to create an alarm for any new HBAs that are added to the environment. As can be seen in the Alarm Policy screenshot below, using VirtualWisdom, an alarm can be easily set up for any HBA that violates the assigned Queue Depth policy.

Alarm Policy: Configuration – HBA Queue Depth Policy Violation

Probe Type Group by
 Metric Set Filter

	Metric Type	Operator	Threshold	Freq	Time Period	Severity	Actions
Trigger	Max Pending Exchanges	>	8	1	1	Minor	Email or SNMP
Re-arm		!=	-1	86,400	86,400	Normal	Internal

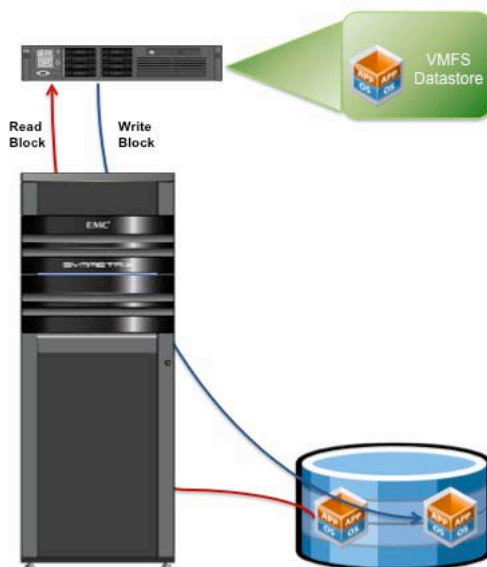
Once this is established, the VirtualWisdom dashboard can be then be used to ensure that the combined Pending Exchanges from all of the HBAs are well balanced across the array and SAN fabric.

Balance I/O Traffic to Ensure Successful Storage vMotion Operations

Initially known as DMotion and only a command line option, Storage vMotion is now far more commonly used in its GUI form. A very useful feature, Storage vMotion enables datastore migrations to take place without any downtime. The advantages of this are huge, but with such disk-intensive operations there can be a significant impact on the SAN fabric and the VM I/O traffic. Performance complaints related to this are that Storage vMotion becomes very slow, or at worst fails, often due to the sudden bandwidth constraints being imposed upon both the source and destination. To counter this and to be best prepared for Storage vMotion operations, it is essential to have clear insight of your SAN utilization and to ensure that I/O traffic is correctly balanced. When a Storage vMotion operation usurps the I/O bandwidth of a shared storage device and subsequently causes other VMs sharing the device to suffer from resource constraints is another example of how a very useful Sphere feature can become a cause of poor performance due to lack of visibility.

Getting the Most from VAAI Primitives

There are three primitives to vSphere API Array integration (VAAI). The first is the Full Copy primitive. With vSphere, when copying data occurs, whether via VM cloning or Storage vMotion, the ESX server becomes responsible for reading every single block. This has to be copied and then written back to the new location. This adds a fundamental load on the ESX server. So for example, when deploying a 100GB VM from a template, the entire 100 GB will have to be read by the vSphere host and then subsequently written, requiring a total of 200 GB of I/O.

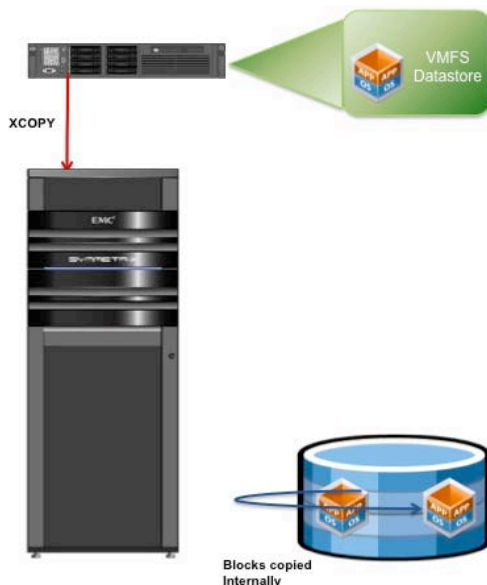


Cases for the Full Copy Primitive:

- vCenter VM cloning** - makes a full copy of VMs and attached virtual disks.
- Storage vMotion** - moves VMs and attached virtual disks from one datastore to another while the VMs are running.

Without VAAI the ESX server has to read every block of the virtual disk to be cloned / moved and then write it back out to the new location

Instead of this host intensive process, the Full Copy primitive operates by issuing a single SCSI command called the XCOPY. The XCOPY is sent for a contiguous set of blocks which in essence is a command to the storage to do the copying of each



Benefits of the Full Copy Primitive:

- Reduces the use of compute and network resources for the copy operation.
- Copy and move operations complete faster.

The ESX server sends a single SCSI command (XCOPY) for a set of contiguous blocks, telling the storage to copy the blocks from one logical block address (LBA), to another LBA.

blocks from one logical block address to another. A significant load is taken off the ESX server and instead applied to the storage array, which can more than easily deal with the copy operations resulting in very little I/O between the host and storage system.

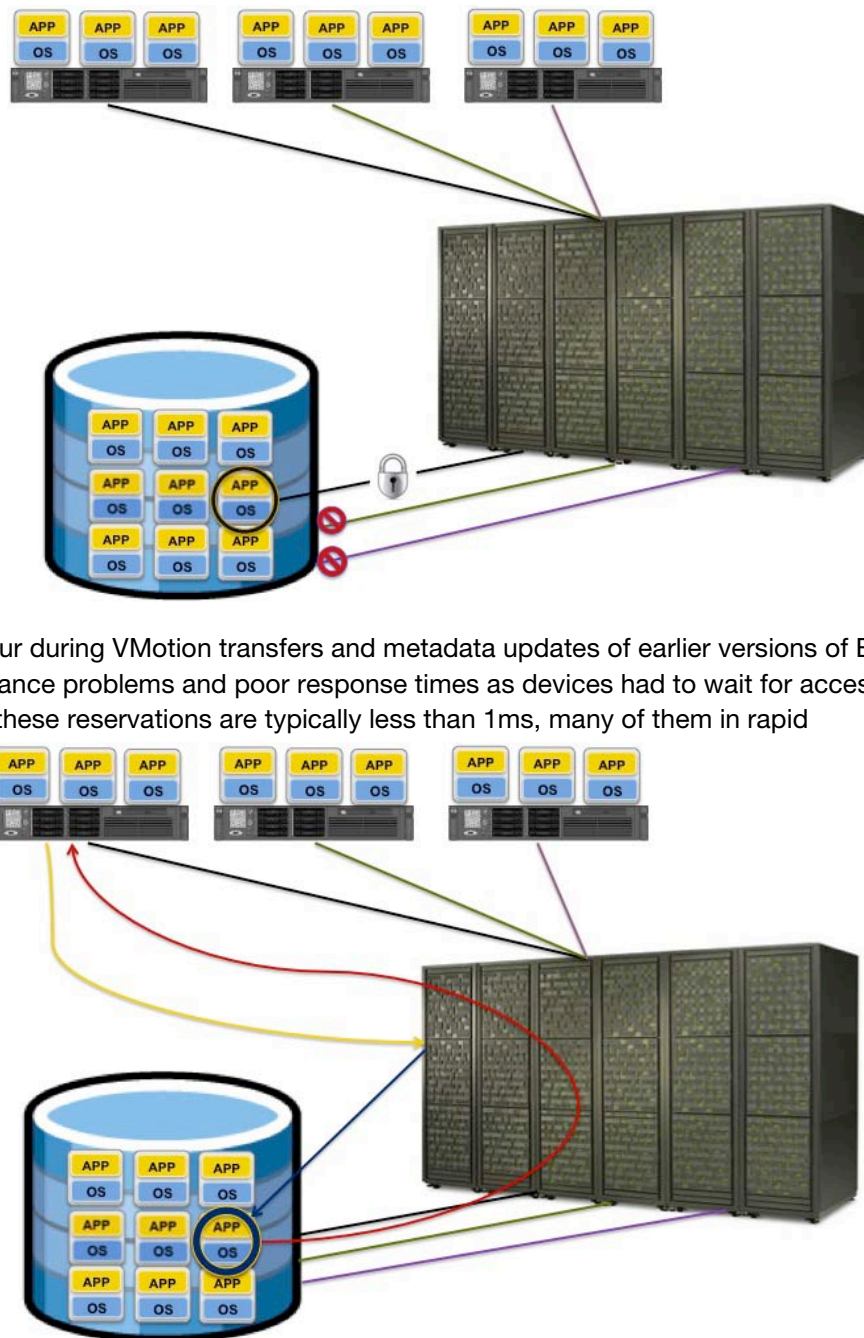
The second primitive, Block Zeroing, is also related to the virtual machine cloning process which, in essence, is merely a file copy process. When a VM is copied from one datastore to another it would copy all of the files that make up that VM. So for a 100 GB virtual disk file with only 5 GB of data, there would be blocks that are full as well as empty ones with free space, i.e. where data is yet to be written. Any cloning process would entail not just IOPS for the data but also numerous repetitive SCSI commands to the storage system for each of the empty blocks that make up the virtual disk file.

Block zeroing instead removes the need to send these redundant SCSI commands from the host to storage. By simply informing the storage system which blocks are zeros, the host offloads the work to the storage without having to send commands to zero out every block within the virtual disk.

The third VAAI primitive is named Hardware Lock Assist. The original implementation of VMFS used SCSI reservations to prevent data corruption when several servers shared a LUN. In the diagram at right, we see what typically occurs without VAAI, i.e. SCSI reservation conflicts. As a normal part of the SCSI protocol, SCSI reservations occur to give exclusive access to a LUN so that competing devices do not cause data

corruption. This used to occur during VMotion transfers and metadata updates of earlier versions of ESX and caused serious performance problems and poor response times as devices had to wait for access to the same LUN. Although these reservations are typically less than 1ms, many of them in rapid succession can cause a performance plateau with VMs on that datastore.

Hardware-Assisted Locking is the elimination of this LUN-level locking based on SCSI reservations. Initially, the ESX server will make a first read on the lock. If the lock is free, the server will then send a Compare and Swap command that is

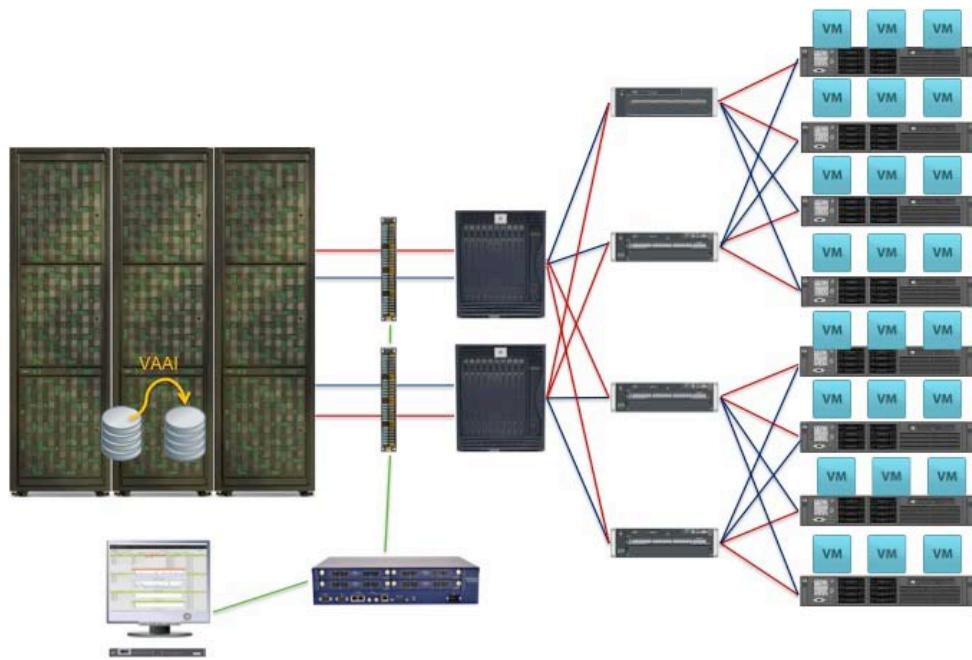


not only the lock data that the server wants to place into the lock but also the original free contents of the lock. The storage system will then read the lock again and compare the current data in the lock to what is in the Compare And Write command. If they are found to be the same, new data will be written into the lock. All of this is treated as a single atomic operation and is applied at the block level (not the LUN level) making parallel VMFS updates possible.

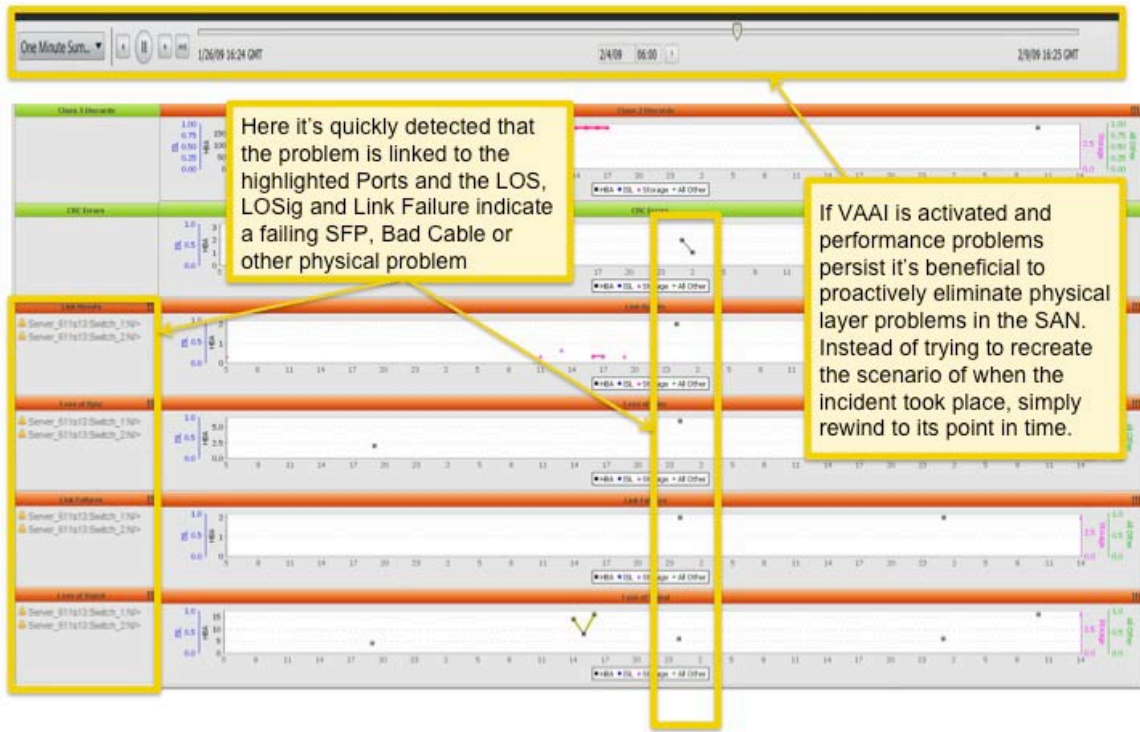
With the benefits brought about by the VAAI primitives and their offloading of tasks onto the storage array, there is now a change and impact upon the SAN fabric's traffic. Moreover, with the Hardware-Assisted Locking Primitive there will be a natural inclination for VM administrators to start deploying larger datastores, as they will be alleviated of the fear of lots of VMs conflicting for LUN access because of locking issues.

VAAI will reduce ESX-to-array traffic and VirtualWisdom can monitor it. VirtualWisdom's capabilities are essential in monitoring the SAN traffic pattern changes that occur, and the potential impact this may have on other applications. This would include monitoring how links and their utilization would change once the primitives of VAAI are initiated. Once these metrics are baselined, it is possible to redesign certain elements

of the SAN fabric such as fan-in and fan-out ratios to make better use of resources. For example, the reduction in SAN traffic caused by VAAI may lead to rethinking how many storage ports and front end directors are required to satisfy application I/O demands. This can lead to significant CAPEX savings as well as optimized performance.



VirtualWisdom's value in gaining the most from VAAI also includes its ability to identify and enable the proactive elimination of physical layer issues which may exist, and consequently hinder the benefits of VAAI. For example, should VAAI be enabled but a physical layer issue exists between the path from host to SAN fabric to storage system, the performance degradation caused by such issues will hinder significantly any benefits that VAAI should achieve. In a 10,000 port environment, identifying such



problems is almost impossible and is akin to trying to find a needle in a haystack. Even worse, this can lead to VAAI being falsely accused of not being effective, when in fact something as trivial as a failed SFP is causing degradation. Looking at this dashboard above, we see how VirtualWisdom can identify and alert the admin of such issues before they cause any noticeable performance degradation. In this example, VAAI has been enabled but it's failing to show any significant performance improvement. Using VirtualWisdom's ability of historical trending, the admin can rewind to when performance issues were still being noticed. In this scenario, we immediately identify that the performance degradation is being caused by a Loss of Sync that is indicative of a speed mismatch between the two identified links, possibly caused by an auto-negotiated-down port. This has led to a Loss of Signal. Had the admin customized an alarm for such issues, VirtualWisdom could have enabled proactive elimination of the problem and subsequently absolved VAAI from any false blame.

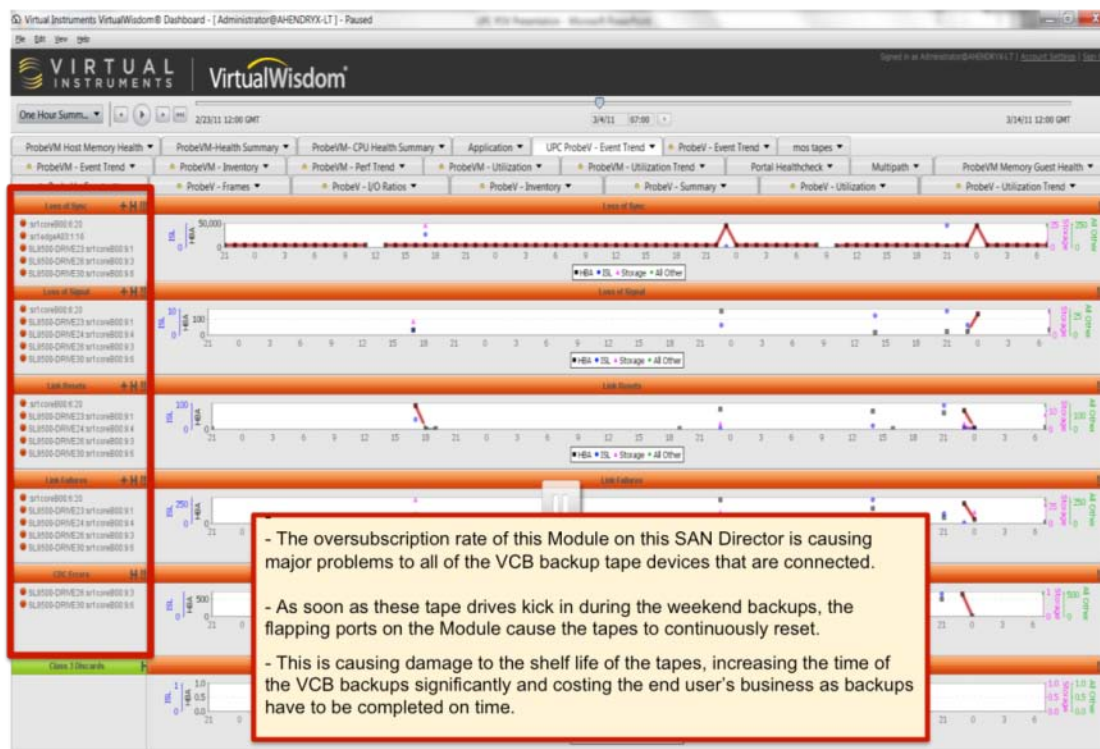
On a wider scale, it can be seen that VAAI is transforming the role of the storage system from a monolithic box of capacity to an off-loadable virtual resource of performance optimized storage processors for ESX servers. For this large virtual pool of resources that now exists between the ESX servers and the storage system to be truly successful, it is essential that the network that interconnects them, i.e. the SAN fabric, is monitored and optimized.

Lack of Isolation, Consequent Effects and Performance Degradation

Another common vSphere/storage performance issue is caused by the Datastore or VM's RDM LUNs being provisioned on very busy RAID groups or aligned to poorly balanced back-end processors on the storage system. This can partly be prevented by using the native storage tool and monitoring the utilization of the backend processors and RAID groups. Such issues are hard to monitor and deal with if due to a forced lack of isolation, a poorly performing application, or jobs that affect other virtualized

applications on corresponding LUNs. A typical example would be that of a nightly backup that has overrun its schedule and then begins to impact the performance of production VMs in the morning hours of the business.

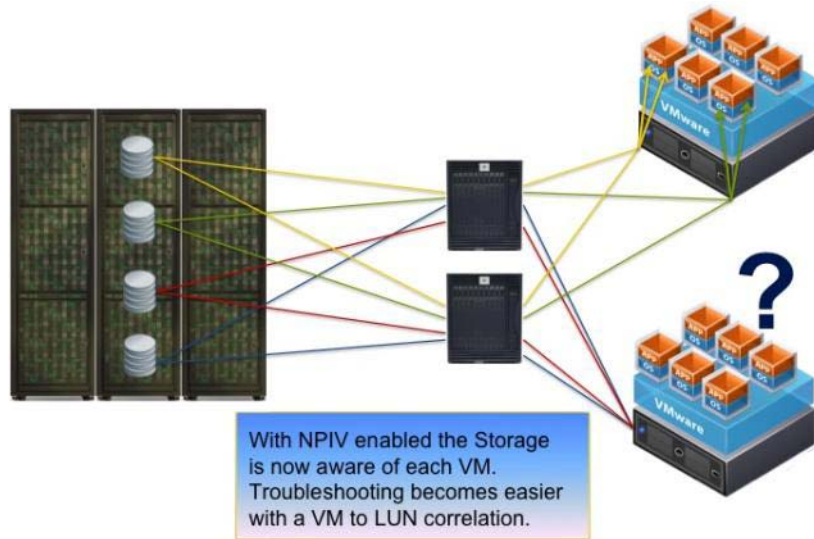
Such situations can be proactively prevented with VirtualWisdom, as in the example below. Here, the backup time has been extended. But it has been proactively identified that the SAN switch ports attached to the LTO4 tape drives are momentarily resetting at different intervals during the night. This has caused the tape drives to reset and prolonged the backup, which is now affecting the Virtual Machines and their corresponding LUNs that share the same RAID Groups. By being able to proactively identify and remediate such issues, their effect on VM performance can immediately be eliminated.



Correlating Issues from the VM to the LUN

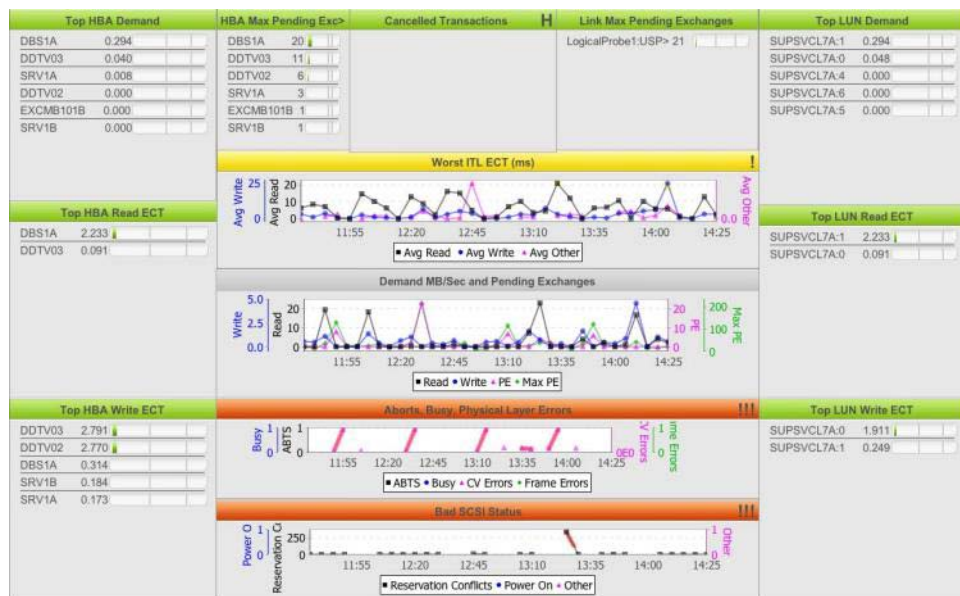
With the abstraction brought about by virtualization it is now more complex to correlate VM to LUN issues. In the case of RDMS it is possible to alleviate this by deploying N_Port ID Virtualization (NPIV) with VirtualWisdom. To elaborate, in Fibre Channel, an N_Port is an end node port on the Fibre Channel fabric i.e. an HBA (Host Bus Adapter) or a target port on the storage. Typically N_Ports have a single N_Port_ID which is assigned by the Fibre Channel switch. This is not to be confused with World Wide Port Name despite the fact that normally the WWPN and the NPort ID have a one to one relationship. NPIV's benefit is that it

presents a single physical N_Port with multiple WWPNs and consequently multiple N_Port_IDs. Once registered, these new WWPNs can be used in exactly the same way as physical ones. Essentially this means that a VM can now be allocated and independently zoned its own WWPN. With VirtualWisdom and NPIV it's now possible to have a VM to LUN correlation as can be seen in the diagram at right.



As mentioned before, with the majority of vSphere deployments using VMFS as opposed to RDM, the NPIV issue in these cases is not applicable. Instead, though using VirtualWisdom, even without NPIV, it is still possible to pinpoint a VM to LUN correlation by a process of elimination. Simply by looking at which LUN is suffering from latency and which VM is responsible for the traffic, such correlations are fairly straightforward to identify. Of course, for 100% accuracy and a quicker time to detect the correlation, NPIV is the easiest route. Looking at the screenshot below, with VirtualWisdom you finally have a VM to LUN correlation, with VMs and their correlating applications on the left hand side leading

all the way to their corresponding LUNs on the right hand side. This significantly aids troubleshooting. The ability to track fabric and storage utilization on an individual VM basis helps tier 1 applications to be virtualized, secure and easily manageable.




Conclusion

For vSphere managers, VirtualWisdom adds SAN I/O intelligence for comprehensive performance optimization and troubleshooting, enabling a more aggressive deployment of virtual machines based on real-time measurements and feedback of I/O performance. By indentifying virtualized application performance bottlenecks, VirtualWisdom results in significantly higher virtual infrastructure utilization and helps administrators deliver on the promise of reduced capital and operational costs and improved business agility promised by server virtualization.

Typical IT management tools cannot provide the sophisticated diagnosis and prevention capabilities necessary to maintain today's complex, heterogeneous, fast-changing environments. Application and virtualization monitoring tools are adequate for optimizing the server environment, but sorely lack I/O subsystem monitoring and analysis capabilities and often can't be used to find the root cause of performance bottlenecks. Because the biggest cause of application latency is in I/O, these tools don't offer the capability to effectively mitigate the risks of running mission-critical applications in a virtual server environment.

Good I/O performance is the most critical component to superior application performance. The key to achieving good I/O performance is deploying instrumentation technology that directly measures what is going on at a deep level in the transaction workflow, from the virtual machine all the way to the LUN on the storage system. After the virtual infrastructure is instrumented and measured, Virtual Instruments makes it easy to analyze the data for optimal performance, availability and resource utilization.

	Corporate Headquarters 25 Metro Drive Suite 400 San Jose, CA 95110 Phone: 408-579-4000 Fax: 408-579-4001	Sales sales@virtualinstruments.com Phone: 408-579-4081	Support support@virtualinstruments.com
--	---	---	--

©2011 Virtual Instruments. All rights reserved. Features and specifications are subject to change without notice. VirtualWisdom, Virtual Instruments, SANInsight are trademarks or registered trademarks in the United States and/or in other countries. All other brands, products, or service names are or may be trademarks or servicemarks of, and are used to identify, products or services of their respective owners. 08/22/11_D