

Created by [drummonds](#) 

Using vscsiStats for Storage Performance Analysis

VERSION 6
PUBLISHED

Created on: May 29, 2009 3:28 PM by drummonds - Last Modified: Oct 21, 2009 4:32 PM by drummonds

Introduction

esxstop is a great tool for performance analysis of all types. However, with only latency and throughput statistics, esxstop will not provide the full picture of the storage profile. Furthermore, esxstop only provides latency numbers for Fibre Channel and iSCSI storage. Latency analysis of NFS traffic is not possible with esxstop.

Since ESX 3.5, VMware has provided a tool specifically for profiling storage: vscsiStats. vscsiStats collects and reports counters on storage activity. Its data is collected at the virtual SCSI device level in the kernel. This means that results are reported per VMDK (or RDM) irrespective of the underlying storage protocol. The following data are reported in histogram form:

- IO size
- Seek distance
- Outstanding IOs
- Latency (in microseconds)
- More!

Running vscsiStats

vscsiStats collection and analysis requires two steps:

1. Start statistics collection.
2. View accrued statistics.

Documentation on command-line parameters are available when running `'/usr/lib/vmware/bin/vscsiStats -h'`.

Starting and Stopping vscsiStats Collection

The tool is started with the following command:

```
/usr/lib/vmware/bin/vscsiStats -s -w <world_group_id>
```

This command starts the process that will accrue statistics. The world group ID must be set to a running virtual machine. The running VMs' IDs can be obtained by running `'/usr/lib/vmware/bin/vscsiStats -l'`.

After about 30 minutes vscsiStats will stop running. If the analysis is needed for a longer period, the start command should be repeated above in this window. That will defer the timeout and termination by another 30 minutes.

Since results are accrued and reported out in summary, the histograms will include data since collection was started. To reset all counters to zero, run `'/usr/lib/vmware/bin/vscsiStats -r'`.

Viewing Statistics

Counters are displayed by using the following command:

```
/usr/lib/vmware/bin/vscsiStats -p <histo_type> [-c]
```

The histogram type is used to specify either all of the statistics or one group of them. Options include all, ioLength, seekDistance, outstandingIOs, latency, interarrival.

Results can be produced in a more compact comma-delimited list by adding the optional "-c" above.

Using vscsiStats Results

Use Case 1: Identifying Sequential IO

Storage arrays can process sequential IO much faster than random IO. You can therefore improve the performance of a sequential workload by placing it on a dedicated LUN to allow the array to optimize access. vscsiStats can help you identify

Search VMware Communities

Performance & VMmark ▾

[Search](#)[Advanced Search](#)

Actions

[View as PDF](#)

Incoming Links

[Re: NFS performance statistics on ESX](#)[Performance Troubleshooting for VMware vSphere 4 and ESX 4.0](#)[Re: I can't see disk performance on all of my VMs](#)[Re: Hidden tool in ESX to obtain vSCSI stats.](#)[Re: NFS Performance Analysis](#)

More by drummonds

[Storage Queues and Performance](#)[Advanced Networking Performance Options](#)[Understanding Performance](#)[vCenter Performance Counters](#)[ESX Monitor Modes](#)[View drummonds's profile](#)

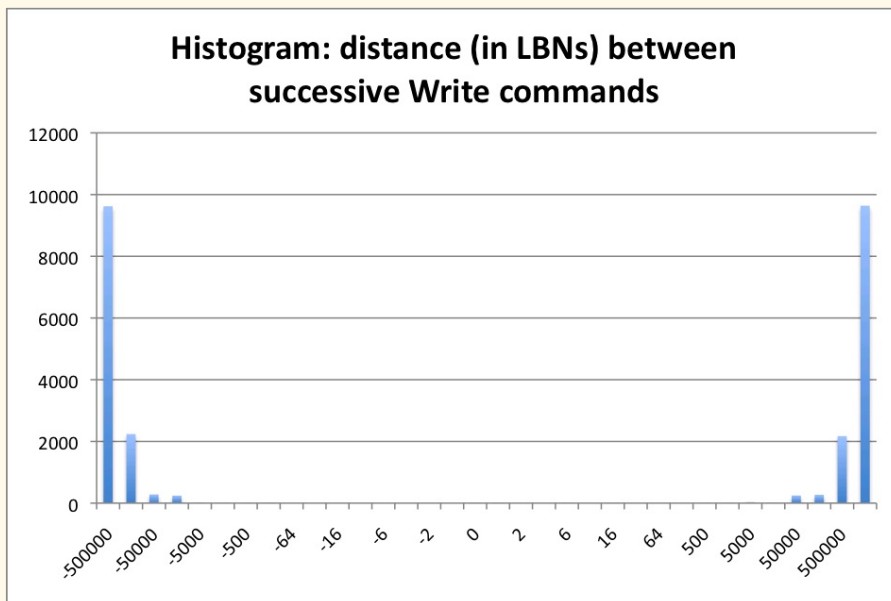
Community Hardware Software

View reported configurations or report your own.

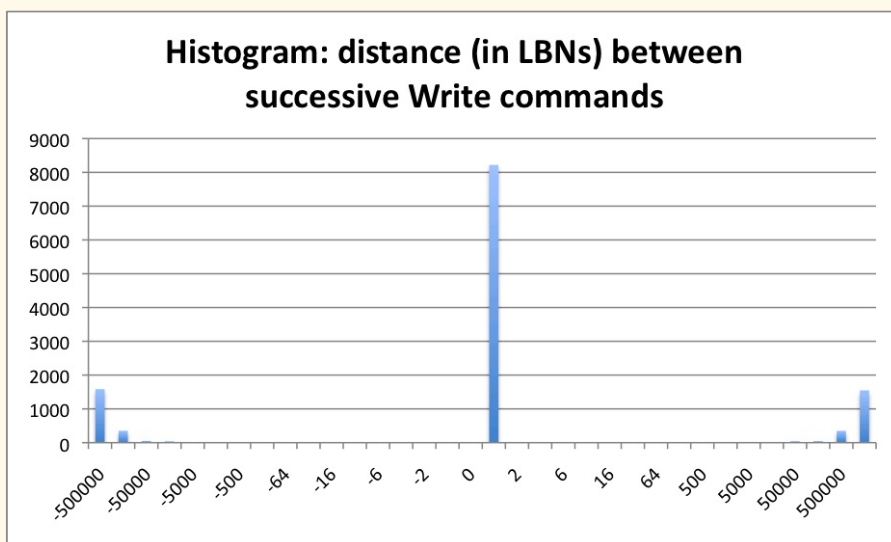
[Learn More](#)

your sequential workloads even if you don't understand anything about the application in the VM.

Take the following graph as example, which I generated by running `'usr/lib/mware/bin/vscsiStats -p seekDistance'`:



This graph shows that most of the commands are being issued a great distance from the previous command. It looks like all of the commands were 50,000 or more logical blocks away from the previous command. When I looked at the raw data, I saw that over 99% of the commands were more than 128 blocks away from the previous command. That's random access if I've ever seen it. Here's the opposite example:

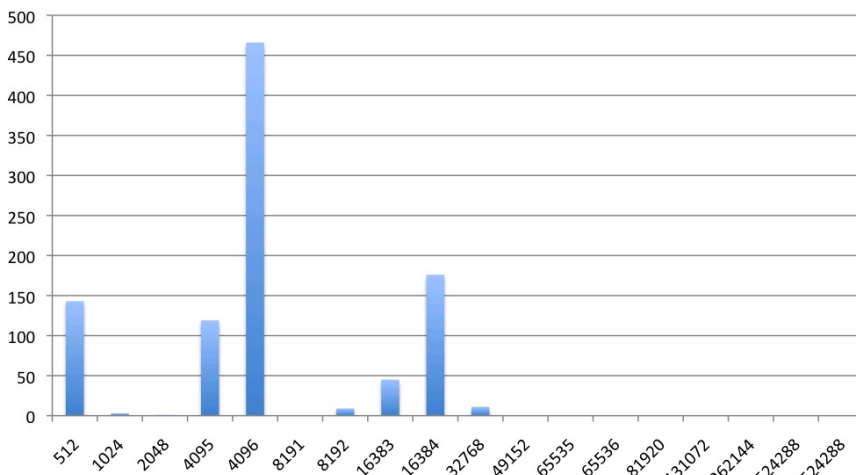


In this case the logical block number (LBN) of each command is most frequently exactly one larger than the previous command. That's the signature of a heavily sequential workload. It shouldn't surprise you to learn that both of these profiles were generated by Iometer using random and sequential writes, respectively.

Use Case 2: Optimizing for IO Sizes

The IO size is an important characteristic of storage profiles. A variety of best practices have been provided by storage vendors to enable customers to tune their storage to a particular IO size. As an example, it may make sense to optimize an array's stripe size to its average IO size. `vscsiStats` can provide a histogram of IO sizes to help this process. The following graph was generated by `'usr/lib/mware/bin/vscsiStats -p ioLength'`:

Histogram: IO lengths of commands



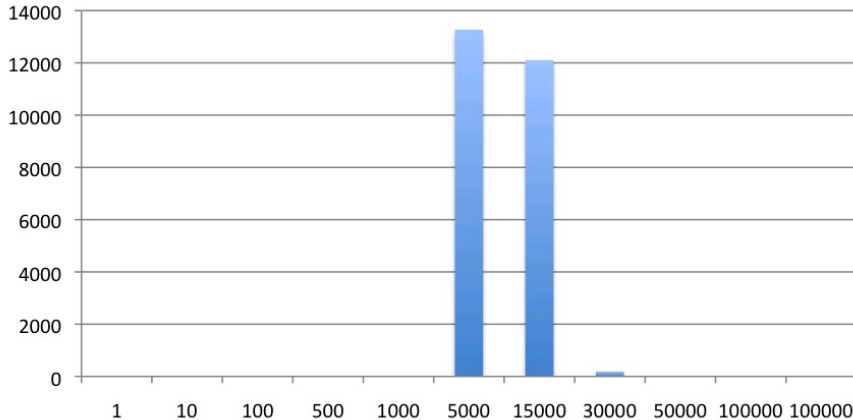
From these results I can see that about a quarter of the commands came in IOs smaller than 4k. About half of the commands were sized to 4k commands. The minute number of remaining IOs were larger than 4k. This signature is common of a VMDK formatted to 4k blocks and supporting OS and application execution. The storage array should be optimized for 4k blocks if this disk's performance is a priority.

Use Case 3: Storage Latency Analysis (Including NFS!)

esxtop is a terrific tool for latency-based storage analysis. Fibre Channel and iSCSI HBAs have device and kernel latencies in esxtop's storage panel. Software iSCSI initiators will show up as vmhba32 (ESX 3.5 and earlier) and vmhba33 (ESX 4.0 and later.) But esxtop does not provide latency statistics for NFS stores.

Because vscsiStats collects its results where the guest interacts with the hypervisor, it is unaware of the storage implementation. Latency statistics can be collected for all storage configurations with this tool.

**Histogram: latency of Write IOs in
Microseconds (us)**



The above graph shows that the server in my office with a single direct-attached SCSI disk is performing as I would expect. About half of all the operations are completing in under 5 ms. The other half take 5-15 ms to complete. A few commands took longer than 15 ms, but the number is so small that it doesn't concern me. Similar results can be seen with NFS arrays.

vscsiStats on ESXi

vscsiStats can be installed on ESXi hosts after putting the host into tech support mode. More information on this process is available on [Scott's blog on the subject on vPivot](#).

Additional Resources

My colleagues Ajay Gulati, Chethan Kumar, and Irfan Ahmad presented at VPACT 09 [Storage Workload Characterization and Consolidation in Virtualized Environments](#). This paper serves as an excellent example of vscsiStats in action.

I learned vscsiStats by reviewing Irfan's VMworld 2007 presentation ([vscsiStats: Fast and Easy Disk Workload Characterization on VMware ESX Server](#)) and playing with the tool. Check out his presentation if you'd like more detail.

Tags: [esxi](#), [vscsistats](#), [analysis](#), [esx](#), [iscsi](#), [nfs](#), [storage](#), [san](#), [nas](#), [disk](#), [vmfs](#)

Comments (31)

irfan.ahmad  says:

Aug 24, 2009 10:35 AM

In case folks are interested in learning more about how this stuff works under the hood, take a look at my related [blog posts](#) and read the IISWC academic [paper](#) on the topic as well.

Finally, Scott is planning on covering vscsiStats usage in detail during his performance labs at VMworld 2009.

Irfan


aenagy says:

Sep 13, 2009 7:32 AM

Irfan:

You mentioned at VMworld 2009 that you would like to see vscsiStats included in vCenter. Do you know if this is going to happen? Soon? If not soon, then will there be an ESXi version? My recommendation to roll-out ESX 4.0 vs. ESXi 4.0 would be influenced by the availability of vscsiStats-like data gathering.

Thanks.

irfan.ahmad  says: in response to: **aenagy**
aenagy: Can you describe your use case a little bit please?

Sep 13, 2009 4:21 PM

Irfan

aenagy says: in response to: **irfan.ahmad**

Sep 13, 2009 5:13 PM

Irfan:

I am getting ready to roll out vSphere and still debating ESX vs. ESXi. There is no problem at this time, I just want to be prepared if we encounter performance problems.

irfan.ahmad  says: in response to: **aenagy**

Sep 17, 2009 1:58 PM

aenagy: please get in touch with drummonds (author of this communities doc) if you need access to vscsiStats on ESXi. Stopping deployment of ESXi should not be gated on this issue.

aenagy says: in response to: **irfan.ahmad**

Sep 28, 2009 7:06 AM

Will do.

Thanks.

lamw  says: in response to: **irfan.ahmad**

Oct 6, 2009 10:48 PM

Is vscsiStats actually available for ESXi? I was under the impression it was only available on Service Console with the classic ESX

=====

William Lam

VMware vExpert 2009

VMware ESX/ESXi scripts and resources at: <http://engineering.ucsb.edu/~duonglt/vmware/>

[vGhetto Script Repository](#)


[VMware Code Central - Scripts/Sample code for Developers and Administrators](#)

[VMware Developer Community](#)

Twitter: [@lamw](#)



If you find this information useful, please award points for "correct" or "helpful".

irfan.ahmad  says: in response to: **lamw**

Oct 7, 2009 9:37 AM

lamw: see my response to aenagy of Sep 17.

drummonds  says: in response to: **lamw**

Oct 26, 2009 10:56 PM

I think you have seen my blog about this already, but I have updated this document to answer your question.

Scott

More information on my blog and on Twitter:

nkiesel says:

Dec 18, 2009 5:41 PM

How can I link the worldgroupIDs and virtual disk handleIDs to objects in the rest of the world? I could of course find e.g. the VMuuid by doing a lookup using SOAP for that machine, but what about the disks if e.g. a VM uses more than one disk?

irfan.ahmad  says: in response to: **nkiesel**

Dec 18, 2009 5:44 PM

try vscsiStats -l and let me know if that helped at all. That's -l as in 'dash' 'el'.

Irfan

nkiesel says: in response to: **irfan.ahmad**

Dec 18, 2009 6:28 PM

Not really. What I see for e.g. a virtual machine that uses 2 hard disks (one on iSCSI and one on NFS) is:
/usr/lib/vmware/bin/vscsiStats -l
Virtual Machine worldGroupID: 1144, Virtual Machine DisplayName: Probe {
Virtual SCSI Disk handleID: 8200
Virtual SCSI Disk handleID: 8210
}

The histograms refer to the worldGroupID and handleID, but how do I know if 8200 is the iSCSI or the NFS disk?

irfan.ahmad  says: in response to: **nkiesel**

Dec 18, 2009 7:20 PM

The virtual disks should be presented to the VM in list order of that output. So, Hard Disk 1 should be the first one presented and so forth.

Irfan

nkiesel says: in response to: **irfan.ahmad**

Dec 18, 2009 8:02 PM

Great! So I could do the following to find this:

- 1) find the hardware for the VM (e.g. via /mob/?moid=3904&doPath=config.hardware)
- 2) iterate over the virtual devices and find those of type VirtualDisk and look at their deviceInfo.label (e.g. "Hard Disk 1")
- 3) for these devices look at the backing.datastore to find out on which datastore they are located

and then map the datastores to the handleIDs in the same order.

Works for me as long as users can't change the labels of the devices (can they)? Or should I rely on the key attribute instead (first disk always seems to have 2000, next one 2001 etc).


Thanks for the help and have a great weekend!

irfan.ahmad  says: in response to: **nkiesel**

Dec 19, 2009 3:20 PM

Sorry, I'm not a scripting expert. Maybe someone else can chip in to help. If possible, I'd like to get a hold of the final script you write though for future reference.

Irfan


lamw  says: in response to: **nkiesel**

Dec 19, 2009 9:05 PM

If what @irfan.ahmad says is true about the mapping of vscsiStats output to VMDK order in the VM configuration, then as @nkiesel pointed out this can easily be retrieved using the VMSphere API.

@nkiesel, There is a unique key for each Device that is kept track on a per VM basis, hence you may see repetitive values such as 2000, 2001, etc. as VMDK(s) afair starts at 2000 and increments by 1 and actually the users can change the labels to whatever their heart desires, if the order is kept upon the creation of the devices then it should be trivial to map.

Here is a script I wrote awhile back that retrieves some info about VMs disk, I've added an additional attribute to capture the VMDK SCSI adapter and target value (SCSI X:Y).

Take a look here for more details:  [getVMDiskInfo.pl](#)

Here's an example:

script output:

```
[vi-admin@scofield ~] $ ./getVMDiskInfo.pl --server reflex.primp-industries.com --username pr  
esxi4-1  
Hard disk 1 = [himalaya-local-SATA.RE4-GP:Storage] esxi4-1/esxi4-1.vmdk  
Hard disk 1 = isThinProvisioned  
Hard disk 1 = persistent  
Hard disk 1 = VirtualDiskFlatVer2BackingInfo  
Hard disk 1 = SCSI (0:0)  
  
Hard disk 2 = [himalaya-local-SATA.RE4-GP:Storage] esxi4-1/esxi4-1_1.vmdk  
Hard disk 2 = isThinProvisioned
```

```

Hard disk 2 = persistent
Hard disk 2 = VirtualDiskFlatVer2BackingInfo
Hard disk 2 = SCSI(3:13)

Hard disk 3 = [himalaya-local-SATA.RE4-GP:Storage] esxi4-1/esxi4-1_2.vmdk
Hard disk 3 = isThinProvisioned
Hard disk 3 = persistent
Hard disk 3 = VirtualDiskFlatVer2BackingInfo
Hard disk 3 = SCSI(3:2)

Hard disk 4 = [himalaya-local-SATA.RE4-GP:Storage] esxi4-1/esxi4-1_3.vmdk
Hard disk 4 = isThinProvisioned
Hard disk 4 = persistent
Hard disk 4 = VirtualDiskFlatVer2BackingInfo
Hard disk 4 = SCSI(0:4)

Hard disk 5 = [himalaya-local-SATA.RE4-GP:Storage] esxi4-1/esxi4-1_4.vmdk
Hard disk 5 = isThinProvisioned
Hard disk 5 = persistent
Hard disk 5 = VirtualDiskFlatVer2BackingInfo
Hard disk 5 = SCSI(2:5)

```

vscsiStats output:

```

[root@himalaya ~]# /usr/lib/vmware/bin/vscsiStats -l

.....

Virtual Machine worldGroupID: 40133, Virtual Machine Display Name: esxi4-1 {
  Virtual SCSI Disk handleID: 8578
  Virtual SCSI Disk handleID: 8579
  Virtual SCSI Disk handleID: 8580
  Virtual SCSI Disk handleID: 8581
  Virtual SCSI Disk handleID: 8582
}

```

From this output, you should be able to map the vSCSI handleID which are just counters that are initiate for each disk and will change as a VM is powered on/off, similar to worldGroupID.

I haven't done extensively testing, but you should be able to validate this by adding load on certain disks and see if it maps back as @irfan.ahmad said


William Lam


VMware vExpert 2009

VMware ESX/ESXi scripts and resources at: <http://engineering.ucsb.edu/~duonglt/vmware/>

Twitter: @lamw

vGhetto Script Repository

 [Getting Started with the vMA\(tips/tricks\)](#)

 [Getting Started with the vSphere SDK for Perl](#)

[VMware Code Central - Scripts/Sample code for Developers and Administrators](#)

[VMware Developer Community](#)



If you find this information useful, please award points for "correct" or "helpful".

nkiesel says: in response to: **lamw**


Dec 21, 2009 11:16 AM

Can I rely on the fact the the vSCSI handleIDs only change at VMboot time. What about hot-swapping (e.g. are disks added to a VM while it's running visible/accessible to the VM before the next reboot)?

Just curious: is there any other place these vSCSI handleIDs (or the worldGroupIDs) are used?

I will of course test all this in my lab...

</nk>

irfan.ahmad  says: in response to: **nkiesel**

Dec 22, 2009 10:57 AM

All virtual disks have vscsi handles (and therefore handle IDs).

If you suspend/resume, vmotion/svmotion or initiate any other such operations that require registration of virtual disks, of course, the IDs will change.

These IDs are really just opaque identifiers to uniquely identify virtual disks and you should not infer any semantics from them since the semantics are likely to change in the future.

Irfan

irfan.ahmad  says: in response to: **irfan.ahmad**

Dec 22, 2009 11:17 AM

FWIW: I really think that the folks active on this thread should take a good look at <http://communities.vmware.com/docs/DOC-10104>. The paper referenced there and published at VPACT is a great resource for comparing your workloads' vscli stats with those of 5 major benchmarks and also understanding the performance costs and benefits of consolidation.

Internally, we refer to that paper a lot when trying to understand new workloads we observe.

IMHO, that should really be required reading for vSphere storage admins.

Irfan

nkiesel says: in response to: **irfan.ahmad**

Dec 22, 2009 11:22 AM

Yeah, thanks for the clarification. Would still be helpful if there were somewhere an explicit mapping between the vSCSI handleIDs and the VirtualDisk object accessible via mob/SOAP (I mean other than the ordering of the IDs which I'm using right now).

Anyway, thanks a lot for all the comments. Oh, and am I correct to assume that the unit for the ioLength histograms is bytes, and that the "mean" is calculated over the real length of the packets and not from the histogram?

</nk>

irfan.ahmad  says: in response to: **nkiesel**

Dec 22, 2009 11:36 AM

The two things you want connected don't exist in the same universe 😊 Still, the larger point is that this analysis should be easier to do. Ideally this data should be available even through the vCenter API.

If you find this data useful (especially take a look at <http://communities.vmware.com/docs/DOC-10104> for ideas on how far you can stretch this analysis), please consider filing a feature request for the plumbing of the stats.

Yes to bytes. Means are calculated online before data is inserted into the histogram. Off to vacation now. Look forward to your comments on the storage workload characterization paper referred to above.

Irfan

nkiesel says: in response to: **irfan.ahmad**

Dec 22, 2009 12:00 PM

Just one last question- hopefully before you go in your (I'm sure well-deserved) vacation: how is the mean calculated? Is that the same as the average (i.e. sum of all packet sizes divided by number of packets?)

</nk>

irfan.ahmad  says: in response to: **nkiesel**

Dec 22, 2009 12:01 PM

Basically that. But of course the online version of that formula.

Irfan

nkiesel says: in response to: **irfan.ahmad**

Jan 12, 2010 5:19 PM

Hi,

I heard rumors that the next version of ESX (ESX4.1 ?) will have NFS stats available via SOAP. Any idea if is that correct?

Best,
Norbert

drummonds  says: in response to: **nkiesel**

Jan 12, 2010 5:42 PM

Is it just coincidence that I had to dodge this question on my blog just an hour ago? 😊

I provided the best answer VMware can give in the comments section to the blog on vscliStats for ESXi:

<http://pivot.com/2009/10/21/vscliStats-for-esxi/#comments>

Scott

More information on my blog and on Twitter:

<http://pivot.com>

<http://twitter.com/drummonds>

nkiesel says: in response to: **drummonds**

Jan 12, 2010 7:26 PM

Hey, don't blame me for trying 😊

Fair enough. Of course the next logical question is: what does it take to get a hand on a development (or beta) version?

Best,
Norbert

drummonds  says: in response to: **nkiesel**
Also answered on vPivot:

Jan 13, 2010 10:29 AM

<http://vpivot.com/2009/10/21/vscsistats-for-esxi/#comments>

The beta program is invite only. Please contact your account team for an invitation.

Scott

More information on my blog and on Twitter:

<http://vpivot.com>

<http://twitter.com/drummonds>

akutz  says: in response to: **irfan.ahmad**

Jan 20, 2010 2:39 PM

I've come up with a way to map the VMDKs to HandleIDs in a conclusive manner (takes time into account):

<http://communities.vmware.com/message/1460390#1460390>

nkiesel says:


Feb 18, 2010 3:55 PM

Hi,

still working with vscsiStats, very nice tool!

I'm currently trying to get some fine-grained statistics, so I (in a loop) reset the stats, sleep for some seconds and then print the histograms. This works fine overall, however I sometimes see vscsiStats errors (error code 0xbad0094, Failed fetch VSI node (hasVmm)) and/or absurdly high numbers (e.g. 1193970 seconds for max IO write latency). Seems that happens more often (or only?) on ESX4i.

Any idea what I'm doing wrong?

akutz  says: in response to: **akutz**

Feb 25, 2010 12:36 PM

I figured out the pattern for the handle IDs.

The virtual SCSI handle Id is determined by looking first at IDE HDs and then SCSI HDs. The handle Ids are contiguous with regards to the x:y adapter:target numbering scheme with IDE taking precedence.

Here is the Python library I wrote to create a map between handle Ids and the VMDK files.

<http://pastie.org/842974>

selected says: in response to: **akutz**

Aug 4, 2010 1:19 AM

Anybody already figured out the relation between a handleid from vscsistats to a vmdk?

I tried the run pastie-842974, but I don't get any information. Maybe someone can help me with this.

Thnx

[▲ Up to Documents in Performance & VMmark](#)

Community Resources

Developer Center
Documentation
Technical Papers
Compatibility Guides

VMworld

VMworld.com
VMworld Session & Labs

Get Support

Downloads & Patches
Product Support Centers
Manage Licenses
Search the KnowledgeBase

Other Links

Contact Us
About VMware
Jobs at VMware
Customer Case Studies
Events