

# Whats in the VirtualCenter database? ...

---



esiebert7625 6,794 posts since

Oct 23, 2006 I did some research on this to better understand how VC uses it database. Below is the results of this if anyone is interested:

**What data is stored in the VirtualCenter database?** # As of version 2.0.1 of VirtualCenter the database consists of the following tables. The database mainly consists of alarm/event data, HA/DRS data, ESX host information, task/scheduled tasks and VM information. All ESX server and VM configuration data is stored on each ESX server and is simply read and displayed by VC. VC is just a central management front end to set configuration information and displays information that is read from all ESX hosts instead of having to manage each server individually. You can also use the VIC to connect directly to the ESX servers without VirtualCenter and modify the same configuration data. Once you add a ESX host back into VC it reads all the configuration info from that host.

The database is not critical to the operation of ESX servers or their virtual machines, they would continue to function normally if VC or it#s database were unavailable (Except for DRS and vMotion which would not work, HA would still work). If the database were to crash and a new one created you could add your ESX servers back in and it would repopulate the configuration information. The only data unique to the database is performance statistics, alarms, events, tasks, resource pools and custom attributes. This is not official documentation and is based on my browsing the database schema and data with a SQL client. Using a SQL client like Toad (Quest Software) or WinSQL Lite (Free) you can browse the data contained in these tables, you can also write SQL code to query information contained in the tables. Below is some sample SQL code to query information on VM#s and hosts. For information on the views defined in VirtualCenter see this white paper: [http://www.vmware.com/pdf/vc\\_dbviews\\_2x.pdf](http://www.vmware.com/pdf/vc_dbviews_2x.pdf)

**VPX\_ACCESS** # Used to store VC users and groups. This table has 5 columns and as many rows as you have users & groups defined in VC.

**VPX\_ALARM** # Used to store VC alarm definitions. This table 10 has columns and as many rows as you have defined alarm definitions.

**VPX\_ALARM\_ACTION** # Used to store VC alarm actions and triggers. This table has 10 columns and two rows for every defined alarm definition.

**VPX\_ALARM\_EXPRESSION** # Used to store alarm conditions and expressions. This table has 9 columns and two rows for every defined alarm definition.

**VPX\_ALARM\_REFRESH** # Has to do with alarms, not sure what it is used for. This table has 3 columns and had no rows in my case.

**VPX\_ALARM\_RUNTIME** # Used to store alarm events for each host or VM that is covered by a defined alarm, this table has 7 columns and as many rows as hosts/VM#s that have alarms configured. For example if you had 10 VM#s and each had 3 alarms assigned to them you would have 30 rows.

**VPX\_COMPUTE\_RESOURCE** # Used to store resource information for DRS, This table has 15 columns and a small amount of rows depending on how many resource pools are defined.

**VPX\_COMPUTE\_RESOURCE\_DAS\_VM** # Used to store resource information for HA (priority, # of failure, retry period, power off/on isolation). This table has of 6 columns and as many rows as you have VM#s that are part of HA.

**VPX\_COMPUTE\_RESOURCE\_DRS\_VM** # Used to store resource information for DRS (enabled, behavior). This table has 4 columns and as many rows as you have VM#s that are part of DRS.

**VPX\_CUSTOMIZATION\_SPEC** - Not sure what it is used for, this is usually a small table consisting of 6 columns and had no rows in my case.

Whats in the VirtualCenter database? ...

**VPX\_DATACENTER** # Used to store Data Center information in VC, This table has 4 columns and as many rows as you have defined Data Centers.

**VPX\_DATASTORE** # Used to store VMFS/NAS datastore information (storage URL, capacity, free space, type). This table has 9 columns and as many rows as you have defined data stores (RDM#s will not show up here).

**VPX\_DS\_ASSIGNMENT** # Used to store what datastores are assigned to each VM (mount path, id, mode). This table has 6 columns and as many rows as each of your VM#s assigned datastores.

**VPX\_ENTITY** # Used to store the names and ID#s of all entities in VC (VM#s, ESX hosts, Folders, Data Centers). This table has 4 columns and as many rows as you have individual entities.

**VPX\_EVENT** # Used to store all events as a result of tasks or alarms in VC (event type, date/time, VM name, username, category, hostname), this table is typically large and has 15 columns and usually a large amount of rows but is generally small in megabytes, 50,000 rows will equal approximately 9MB.

**VPX\_EVENT\_ARG** # This corresponds to the VPX\_EVENT table and contains event ids, argument types & data and miscellaneous IDs. This table is usually pretty large and contains the text of the events from the VPX\_EVENT table. This table has 14 columns and usually has more records then the VPX\_EVENT table, 150,000 records will generally use about 20MB of space.

**VPX\_FIELD\_DEF** # Used to store custom attribute names that are displayed in the VI client. This table has 2 columns, id and name, and as many records as custom attributes that you have defined.

**VPX\_FIELD\_VAL** # Used to store custom attribute values that are displayed in the VI client. This table has 3 columns, field id (corresponds to VPX\_FIELD\_DEF table), entity id (corresponds to VPX\_ENTITY table) and value. It will have as many rows as you have values for custom attributes.

**VPX\_GUEST\_DISK** # Used to store disk space information for VM#s. This table has 4 columns, vm id (corresponds to VPX\_VM table), path (drive letter), disk capacity and free space. It will have a row for each drive partition that a VM has configured.

**VPX\_GUEST\_IP\_ADDRESS** # Used to store IP address information for VM#s. This table has 3 columns, vm id (corresponds to VPX\_VM table), device id (usually 4000, increments if you have more then one IP address), and ip address. It will have a row for each ip address that a VM has configured.

**VPX\_GUEST\_NET\_ADAPTER** # Used to store the VM network name for each network adapter, these correspond to the network names in each vswitch configuration. This table has 5 columns vm id (corresponds to VPX\_VM table), device id (usually 4000, increments if you have more then one IP address), mac address, is connected and network name. It will have a row for each network adapter that a VM has configured.

**VPX\_HIST\_STAT** # Used to store historical performance statistics that are collected by VirtualCenter. This is the biggest table and only has 5 columns (sample id, stat id, entity id, device id and stat value) but can have millions of rows in it.

**VPX\_HOST** # Used to store ESX server host information. This table has 46 columns (host configuration data) and will have a row for each host in VC.

**VPX\_HOST\_CPU** # Used to store ESX server host CPU information. This table has 6 columns (host id, cpu index, hertz, bus hertz, cpu description and cpu vendor) and will have a row for each CPU (not core) that a ESX server has in it.

**VPX\_HOST\_CPUID\_FEATURE** # Used to store ESX server host CPU Identification masks. This table has 7 columns (host id, feature level, feature vendor, EAX mask, EBX mask, ECX mask, EDX mask) and usually has 5 rows for each ESX server that you have regardless of the number of CPUs in each host.

**VPX\_HOST\_CPU\_CPUID\_FEATURE** # Similar to the previous table. This table has 8 columns (same as above plus a cpu index column) and usually has 10 rows per ESX server that you have regardless of the number of CPUs in each host.

**VPX\_HOST\_CPU\_THREAD** # Used to store ESX server host CPU thread information. This table has 3 columns (host id, cpu index, thread id) and a row for each CPU core in each ESX server. If you had a dual-core server there would be 4 rows with thread ID#s of 0,1,2 and 3.

Whats in the VirtualCenter database? ...

**VPX\_HOST\_NODE** # Used to store ESX server host memory information. This table has 4 columns (host id, numa id, mem range begin, mem range length) and a row for each CPU (not core) that a ESX server has in it.

**VPX\_HOST\_NODE\_CPU** # Used to store ESX server host CPU and memory id#s. This table has 3 columns (host id, cpu id, numa id) and a row for each CPU core in each ESX server.

**VPX\_HOST\_PCI\_DEVICE** # Used to store ESX server host hardware information. This table has 12 columns (host id, pci id, class id, bus, slot, pci function, vendor id, sub vendor id, vendor name, device id, sub device id, device name) and a row for each hardware device in each ESX server (approx. 32). This includes NIC, FC cards, processors, video cards, USB, etc.

**VPX\_HOST\_VM\_CONFIG\_OPTION** # Unsure what this is used for, as the name implies it has something to do with VM configs. This table has 4 columns (host id, config option ver, data, array index) and 2 rows for each ESX server.

**VPX\_LICENSE** # Used to store ESX server license information. This table has 3 columns (serial number, serial key, type) and appears to not be used if you use a License Server in your environment. Possible used if you use host based licenses instead of a license server.

**VPX\_LOCK** - Unsure what this is used for, as the name implies it has something to do with locks. This table only has one column (id) and only one row that has a value of 0.

**VPX\_NETWORK** # Used to store ESX server host network name configurations, these correspond to the network names in each vswitch configuration. This table has 3 columns (id, name, data center id) and a row for each unique network name configuration.

**VPX\_NW\_ASSIGNMENT** # Used to store VM to network name mappings. This table has 2 columns (network id, entity id) and a row for each NIC that every VM has configured.

**VPX\_OBJECT\_TYPE** # Used to store object type names for VirtualCenter (ie. vm, host, alarm, task). This table has 2 columns (id, name) and usually 12 rows.

**VPX\_PARAMETER** # Used to store VirtualCenter configuration parameters (ie. smtp settings, snmp settings, port numbers, time outs). This table has 2 columns (name, value) and approximately 42 rows.

**VPX\_PRIV\_ROLE** # Used to store all VirtualCenter privileges (ie. VirtualMachine.Interact.PowerOn, ScheduledTask.Run) that can be assigned to users and groups. This table has 2 columns (privilege name, role id) and as many rows as there are unique privileges (approximately 293).

**VPX\_RESOURCE\_POOL** # Used to store Resource Pool information. This table has 13 columns (id, config spec, allocated cpu, allocated vm cpu, allocated mem, allocated vm mem, available pool mem, available vm mem, current cpu, current mem, overall status) and as many rows as you have resource pools defined.

**VPX\_ROLE** # Used to store all VirtualCenter Role information (ie. VirtualMachineAdministrator, VirtualMachineUser). This table has 2 columns (id, name) and will have as many rows as roles that are defined.

**VPX\_SAMPLE** # Used to store historical statistic sample times and intervals, corresponds with VPX\_HIST\_STAT table. This table has 3 columns (id, sample time, sample interval) and usually a large amount of rows (although now where near as large as VPX\_HIST\_STAT).

**VPX\_SCHEDULED\_TASK** # Used to store scheduled task information in VirtualCenter. This table has 15 columns and will have as many rows as scheduled tasks that are defined.

**VPX\_SCHED\_ACTION** # Used to store scheduled task action information in VirtualCenter. This table has 3 columns (scheduled task id, action type, action data) and will have as many rows as scheduled tasks that are defined.

**VPX\_SCHED\_SCHEDULER** - Used to store scheduled task scheduler information in VirtualCenter. This table has 16 columns and will have as many rows as scheduled tasks that are defined.

**VPX\_SEQUENCE** - Unsure what this is used for. This table has 2 columns (id, name) and in my case 0 rows.

**VPX\_SNAPSHOT** # Used to store snapshot information for VM#s. This table has 11 columns (id, host snapshot id, vm id, snapshot name, snapshot desc, create time, power state, is quiesced, parent snapshot id, is current snapshot, config) and will have a row for as many snapshots that exist for your VM#s.

Whats in the VirtualCenter database? ...

**VPX\_STAT\_CONFIG** # Used to store statistic collection intervals in VirtualCenter. This table has 3 columns (length, name, sample interval) and as many rows as you have collection intervals defined (usually 4).

**VPX\_STAT\_DEF** # Used to store statistic definitions in VirtualCenter. This table has 7 columns (id, rollup type, name, group name, type, unit, associate ids) and has approximately 168 rows.

**VPX\_TASK** # Used to store all task information (ie. VM Power On, VM Re-configure, Alarm Create) that occurs in VirtualCenter. This table has 27 columns and as many rows as tasks that have occurred in VirtualCenter (can be thousands).

**VPX\_VERSION** # Used to store VirtualCenter database version (ie. VirtualCenter Database 2.0). This table has 2 columns (ver id, version value) and usually one row .

**VPX\_VM** # Used to store virtual machine configuration information (ie. vm id, path to vmx file, uuid, os, state, memory, resource group, ip address, vmware tools version, etc.) read from the ESX server is is hosted on. This table has 41 columns and as many rows as VM#s that are managed by VirtualCenter.

Sample SQL code to query Disk Space info from all VM#s (in gigabytes):

```
o select b.name, path, round(capacity/1073741824,2) "Total", round(free_space/1073741824 ,2) "Free" from vpx_guest_disk a, vpx_entity b where a.vm_id = b.id order by b.name
```

Sample SQL code to display Up Time info from all VM#s:

```
o select b.name, (to_char(sysdate, 'J') - to_char(boot_time, 'J')) "Up Days" from vpx_vm a, vpx_entity b where a.id = b.id order by b.name
```

### **How can I purge old statistic data from the VirtualCenter database?**

The size of the Statistics database (VPX\_HIST\_STAT) will vary based on the number of hosts & VM#s managed, frequency of performance data collection, collection level and type of database. Each stat sample collected is about 60 bytes for SQL, 100 bytes for Oracle, and each event stored is 1600 bytes for SQL, 600 bytes for Oracle.

Using default settings, the statistical data for 25 hosts running 8-16 VMs per host will plateau around 40-60 MB in a year (80-140 MB if set to #full#). Each month, the average number of events generated will also consume about 190 MB in SQL, and 70 MB in Oracle. Total DB size after a year is expected to be around 2.20 Gb in SQL, and 1.0 Gb in Oracle.

Using default settings, the statistical data for 75 hosts running 8-16 VMs per host will plateau around 90-150 MB in a year (200-330 MB if set to #full#). Each month, the average number of events generated will also consume about 190 MB in SQL, and 70 MB in Oracle. Total DB size after a year is expected to be around 2.40 Gb in SQL, and 1.2 Gb in Oracle.

The statistic sample frequency and retention is specified in the VI client. Select Administration, VirtualCenter Management Server Configuration from the top menu then select Statistics. The default settings are:

- o Past Day # 5 minutes per sample, 288 total samples
- o Past Week # 15 minutes per sample, 672 total samples
- o Past Month - 60 minutes per sample, 720 total samples
- o Past Year # 1440 minutes per sample, 365 total samples

For example if you wanted to collect less statistical data you could change the settings to something like below:

- o Past Day # 15 minutes per sample, 96 total samples
- o Past Week # 60 minutes per sample, 168 total samples
- o Past Month - 120 minutes per sample, 360 total samples
- o Past Year # 1440 minutes per sample, 365 total samples

Whats in the VirtualCenter database? ...

Or if you wanted to collect more statistical data you could change the settings to something like below (caution your database will be very large):

- o Past Day # 1 minutes per sample, 1440 total samples
- o Past Week # 5 minutes per sample, 2016 total samples
- o Past Month - 30 minutes per sample, 1440 total samples
- o Past Year # 60 minutes per sample, 8760 total samples

You can also delete any ones you do not want, for example if you only want a weeks worth of data then delete the Past Month and Past Year. Once the total samples number has been reached the older samples are automatically purged from the database. If you modify these values you must follow these rules.

- o The new sampling period must be a multiple of a previous one. In the defaults you'll notice that week (15) is a multiple of day (5), month (60) is a multiple of week(15) and year (1440) is a multiple of month (60).
- o The new sampling length must be longer than a previous one. Week (15) is longer then day (5), etc.
- o The user-specified name of the historical interval must be unique. (In case you add new ones)
- o Note: Existing data is reset (lost) when you change the interval configuration. However, only the data for that interval is reset. For example, if you change only the weekly time interval, the daily and monthly data are retained.

You can also specify the amount of statistical detail that is captured by setting the Statistic Collection Level. The default for this setting is Level 1 which includes basic metrics, it can be changed anywhere up to Level 4 which provides the most statistical detail but can significantly increase the size of the database and can cause additional performance overhead. You can also change the number of threads that are used for collecting performance statistics from managed hosts. Use multiple threads to increase the number of managed hosts and take advantage of the additional processor.

### **How can I purge old Event and Task data from VirtualCenter?**

VirtualCenter does not support doing this directly, the only way you can do this by deleting directly from the database tables. It is strongly recommend to have a functional backup of the VC database before clearing these tables in case of an unforeseen problem. Also the the VC service must be stopped in order to avoid data corruption. This is not a supported procedure by VMware but should be in my opinion, these tables can get pretty large in big environments as every single task and event is preserved in the database.

The tables where the data is stored are VPX\_TASK (Tasks) and VPX\_EVENT, VPX\_EVENT\_ARG (Events). The VPX\_EVENT\_ARG will usually have more rows then the VPX\_EVENT table. This is because the VPX\_EVENT table only has one row per event but the VPX\_EVENT\_ARG table can have multiple rows per event (The link between the 2 tables is the EVENT\_ID column).

- o Shutdown the VirtualCenter service
- o Connect to the database server that is hosting the VC database with a SQL browser/client. You can use a free tool like WinSQL Lite ( <http://www.snapfiles.com/get/winsql.html>) for this.
- o To delete all data in the tables type: #delete from VPX\_TASK# then #delete from VPX\_EVENT# and then #delete from VPX\_EVENT\_ARG# and finally #Commit#
- o Optionally you can do #truncate table VPX\_TASK# then #truncate table VPX\_EVENT# and then #truncate table VPX\_EVENT\_ARG# (Truncate is faster and does not use as much undo space as delete)
- o To selectively delete data older then 30 days: #delete from VPX\_TASK where complete\_time < sysdate - 30)
- o To selectively delete data older then 30 days from VPX\_EVENT and VPX\_EVENT\_ARG is trickier because VPX\_EVENT\_ARG does not have a date field and it tied to the events in VPX\_EVENT by the Event\_id field.
- o First delete from the child table (VPX\_EVENT\_ARG): #delete from vpx\_event\_arg where event\_id in (select a.event\_id from vpx\_event\_arg a, vpx\_event b where a.event\_id = b.event\_id and create\_time < sysdate - 30)

Whats in the VirtualCenter database? ...

- o Next commit the delete by typing #Commit#
- o Then delete from the parent table (VPX\_EVENT): #delete from vpx\_event where create\_time < sysdate # 30#
- o Finally commit again by typing #Commit#
- o Start the VirtualCenter service



[Rob.Bohmann](#) 583 posts since

May 31, 2005 1. **Re: Whats in the VirtualCenter database?** May 23, 2007 3:53 PM

thank you for compiling this!



[is\\_nt\\_admins@...](#) 40 posts since

Sep 21, 2006 2. **Re: Whats in the VirtualCenter database?** May 31, 2007 8:48 AM

The Disk Space Info query gave me some errors. Keep in mind I am not a SQL guy...

The "o" at the beginning gave a #Could not find stored procedure# error.

Also did not like the rounding functions. Gave a #The conversion of the varchar value '6432133120' overflowed an int column. Maximum integer value exceeded# error.

I ended up modifying the query to:

```
select b.name, path, capacity "Total", free_space "Free" from vpx_guest_disk a, vpx_entity b where a.vm_id = b.id  
order by b.name
```

which worked but returned the raw size values. No big deal.

Also, it only seemed to return information about running VM#s.

Thanks anyway, it still took me a long way.



[esiebert7625](#) 6,794 posts since

Oct 23, 2006 3. **Re: Whats in the VirtualCenter database?** May 31, 2007 8:57 AM

👤 in response to: [is\\_nt\\_admins@...](#) No problem, the "o" was actually a bullet from the Word doc I copied it out of and is not part of the query, it looks a little different when pasted into these forums.



[rwg](#) 2 posts since

Jun 23, 2005 4. **Re: Whats in the VirtualCenter database?** Sep 12, 2007 5:57 AM

Great article.

For capacity planing purposes I want to populate an Access database with some of the info in the VC database. This is to allow reservations for planned VM's to be made in our ESX environment without touching the VC database.

A possilbe enhancment for a future releace of VC would be to include some capacity management functionality, taking feedback from ESX performance stats and allowing possible.planned VM's to be 'allocated' - not created but 'reserved'. Does anyone else have ideas on this topic?

What I'm trying to do is write sql to query our VC Oracle database

For each VM i want to list the VM hostname, esx host , each vmkd filename and it's lun location and size.

Whats in the VirtualCenter database? ...

I can't seem to find the table that stores a VM's vmdk filename and size - perhaps this is not stored in the VC database (being stored in the VM's .vmx file).

This Sql code may be useful for anyone else attempting to do a similar task. This is my sql so far - seems to work for Windows VM's , but the VM\_Capacity and VM\_Free\_Space figures for Linux VM's are not accurate.

```
SELECT vpxv_vms.vmid, vpxv_vms.NAME, vpxv_vms.hostid,
vpxv_vm_datastore.ds_id, vpx_datastore.NAME, vpx_datastore.capacity / 1024000000 as Lun_Capacity,
vpx_datastore.free_space / 1024000000 as Lun_Free_Space, vpxv_hosts.NAME, vpx_guest_disk.PATH,
vpx_guest_disk.capacity / 1024000000 as VM_Capacity, vpx_guest_disk.free_space / 1024000000 as
VM_Free_Space
FROM vpxv_vms,
vpxv_vm_datastore,
vpx_datastore,
vpxv_hosts,
vpx_guest_disk
WHERE ( (vpxv_vms.vmid = vpxv_vm_datastore.vmid)
AND (vpxv_vm_datastore.ds_id = vpx_datastore.ID)
AND (vpxv_hosts.hostid = vpxv_vms.hostid)
AND (vpx_guest_disk.vmid = vpxv_vms.vmid)
)
ORDER BY vpxv_vms.NAME ASC , path ASC
```



[jliu](#) 30 posts since

Aug 22, 2005 5. **Re: Whats in the VirtualCenter database?** Oct 26, 2007 12:35 PM

How can I get vmdk size assigned? the propose of that is for charge back. A VM can be allocated for 20GB vmdk, but only 16GB for c\$.

vpx\_guest\_disk looks like the information from guest OS level.

Thanks.



[angoletti1](#) 71 posts since

Dec 2, 2005 6. **Re: Whats in the VirtualCenter database?** Dec 27, 2007 4:56 AM

in response to: [jliu](#)

Hi,

if I try to execute the command:

```
delete from vpx_event_arg where event_id in (select a.event_id from vpx_event_arg a, vpx_event b where
a.event_id = b.event_id and create_time < sysdate - 30)
```

I allways get:

Invalid column name 'sysdate'.

What's the problem?

Whats in the VirtualCenter database? ...

Thanks, Chris



[RParker](#) 5,270 posts since

Dec 6, 2006 7. **Re: Whats in the VirtualCenter database?** Dec 27, 2007 6:29 AM

in response to: [rwg](#)

You need to talk to a SQL admin, you don't need Access.

SQL can handle millions of records from hundreds of thousands of concurrent users, and the VC database is nothing more than a table in a database. A SQL admin can help you copy the table (VC database) to another table and you can manipulate the info on the same SQL database, no need to make a copy. If they know what they are doing, they can even link it (or help) to connect FROM access so you can get this info anytime you want.

There isn't a problem using Access, but it would make things easier if you can have live links to the data you want to copy, just don't change the original table and you are fine.



[mhanbury](#) 80 posts since

Jan 10, 2006 8. **Re: Whats in the VirtualCenter database?** Feb 28, 2008 6:04 AM

Superb post. Thank you for this information.



[shane.presley](#) 73 posts since

Oct 16, 2007 9. **Re: Whats in the VirtualCenter database?** Feb 28, 2008 7:48 AM

Very helpful...I was looking for database size estimates. Thanks!!



[zman8082](#) 19 posts since

Jan 16, 2006 10. **Re: Whats in the VirtualCenter database?** Mar 12, 2008 12:31 PM

This is great information. I'm working on a Web interface so my manager and his manager won't have to ask me everytime they want some information about our VMware environment. The pages are in asp. I have most of it completed, the part on the statistical history will be very useful.

Thanks,

Zman



[bpayne](#) 32 posts since

Aug 3, 2007 11. **Re: Whats in the VirtualCenter database?** Apr 10, 2008 7:07 AM

Great post. We have an issue where our Oracle database is growing at a rapid rate. We've had Vcenter installed and configured for a little over a month now. Our database has already exceeded 11Gb which is way out of line with the estimates given in VCenter statistics. I'm thinking some of this may be due to us giving verbose logging turned on in the beginning when we were having some troubles getting it configured. I'm having our Oracle DBA look at the tables and purge any event and task data to see if that helps.

I'm still concerned there is something else going on as we have changed the logging options yet our tablespace is still growing. We have 4 ESX hosts and 35 VMs (small environment).

Whats in the VirtualCenter database? ...

Does anybody have any ideas as to what might be causing our problem?

Thanks, in advance.



[eric.heilig](#) 22 posts since

Sep 8, 2006 12. **Re: Whats in the VirtualCenter database?** Jul 22, 2008 7:15 AM

Does anybody have similar information specific to VC 2.5? It seems that there have been some major changes with regard to VM performance metrics between 2.0.x and 2.5.

Thanks in advance!

Eric