

VMware vCenter 4.0 Database Performance for Microsoft SQL Server 2008

VMware vSphere 4.0

VMware® vCenter™ Server uses a database to store metadata on the state of a VMware vSphere™ environment. Performance statistics and their associated stored procedure operations are among the largest and the most resource-intensive components of the vCenter database. Hence, the performance of your vCenter database depends upon the frequency at which you collect performance statistics and the level of detail of the statistics you store.

The purpose of this study is to present the performance results of tests we conducted of a vSphere 4.0 environment using Microsoft SQL Server 2008 as the vCenter database. The study measures the amount of time it takes to initiate and run the stored procedures in the vCenter database that periodically perform statistics rollups and purges. By knowing how long these procedures run, you can make informed decisions about what levels to set for the collection of performance statistics, whether you should expect to make allowances for additional hardware memory (or less), and identify what your database storage needs will be regarding the archival of performance statistics data.

VMware vCenter Server Database Overview

vCenter Server uses a database to store and organize all its information about the VMware vSphere environment. The data in the vCenter database can be divided into three broad categories:

- **Inventory**—includes host and virtual machine configurations, resources and virtual machine inventory, user permissions, and roles.
- **Alarms, events, and tasks**—Alarms are notifications that are triggered in response to selected events, conditions, and states that occur with objects in the inventory. Tasks are system activities that do not complete immediately. The task and event data is stored indefinitely in the vCenter database, unless the user specifies the retention period. See “Set Database Retention Policy” on page 8.
- **Performance statistics**—includes the performance and resource utilization statistics for datacenter entities, such as hosts, clusters, resource pools, and virtual machines. vCenter provides configurable settings to control how performance statistics are collected across the vSphere environment. Performance statistics could take up to 90 percent of the vCenter database size and therefore are an important factor in performance and scalability of the vCenter database.

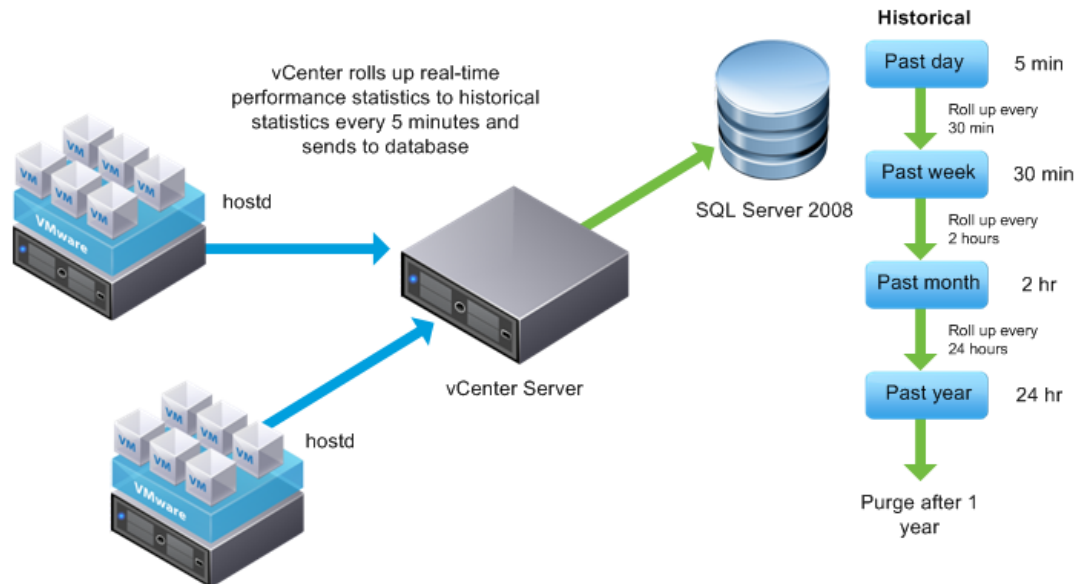
VMware vCenter Server supports a number of databases, which are listed in the vSphere 4.0 *Basic System Administration Guide*.^[1] The information in this study pertains specifically to Microsoft SQL Server 2008, 64-bit.

Performance Statistics Collection in vCenter

VMware vCenter collects real-time statistics from the host agent running on each ESX host. Every 30 minutes, vCenter inserts the real-time statistics into the vCenter database as historical statistics. You can

view the historical statistics for the past day, the past week, the past month, and the past year. As these historical statistics age from day to week to month to year, SQL Server rolls them up into larger granularities. By default, the rolled-up statistics are stored for one year. As shown in Figure 1, the rollup stored procedures run as scheduled jobs in the vCenter database every 30 minutes, every 2 hours, and every 24 hours. You may see spikes in your database's resource utilization when these stored procedures are running.

Figure 1. Stored procedures roll statistics up to the next time interval



You can configure the amount of historical statistics collected in the vCenter database by:

- Changing the interval length or collection frequency
- Changing the collection level
- Enabling or disabling a collection interval

Because the vCenter database size is one of the most important factors in determining the performance and scalability of vCenter, you should configure these settings carefully. For further details, refer to "Working with Performance Statistics" in the *Basic System Administration* guide. [1]

Performance Statistics Database Operations and Their Effects on Performance

The three basic statistics database operations are insert, roll up, and purge. Each operation has its own performance characteristics.

- **Insert**—vCenter inserts performance statistics into the database every 30 minutes. The number of disk I/O writes per second of the log device is the primary factor affecting performance of the insert operation. CPU and memory are secondary factors for insert performance.
- **Roll up**—As performance statistics age, SQL Server rolls them up to the next time interval. The rollup stored procedures run every 30 minutes for day statistics, every two hours for past week statistics, and once a day for past month statistics. Because SQL Server uses parallel processing whenever it can, rollup stored procedures can run three to four times faster in multicore systems provided they have enough memory. SQL Server parallel query is on by default. If rollups seem to be taking a long time, run SQL Trace and compare the CPU time with the total time. If total time is greater than CPU time, the

problem may be insufficient memory allocated to SQL Server, in which case you can improve performance by increasing the amount of memory available to SQL Server.

- **Purge**—After performance statistics have been rolled up and are older than the time period over which they were collected, they are no longer necessary and can be deleted from the database. This regular deletion can leave “holes” in the clustered index and lead to fragmentation. A fill factor of about 70% is recommended for the VPX_HIST_STAT tables. This number means that 30% of each leaf-level page will be left empty, which provides space for index expansion as data is added to the underlying table. The empty space is reserved between the index rows rather than at the end of the index. [2] For details on setting the fill factor, see “Set Fill Factor for Clustered Indexes on Performance Statistics Tables,” page 8.

Performance Tests

We used the following test configuration and procedures to examine the performance improvements for the vCenter database.

Test Environment

Database Hardware

- 2 quad-core Intel Xeon X5355 processors
- 32GB RAM
- 8x 73GB 15K RPM SAS disks in a RAID 0 configuration (local disks)

Database Software

- Operating system: Microsoft Windows Server 2008 Enterprise Edition SP2 64-bit
- Microsoft SQL Server 2008 64-bit

We chose 8 high-speed local SAS disks with RAID 0 for our study. It is also common to use a storage area network (SAN).

Note: vCenter Server also supports the 32-bit version of SQL Server 2008. When you use the 32-bit version, you need to enable AWE. See “Appendix A: Configuring Memory Size for 32-Bit SQL Server” on page 10 for more information on setting memory for 32-bit versions.

Benchmark Setup

60 hosts/600 VMs is the largest inventory size supported in the previous version of vCenter, and 300 hosts/5000 VMs is the largest supported by vCenter 4.0. We selected inventory sizes in the range between the two limits.

Table 1. Number of hosts and virtual machines in test samples

Inventory: 660	Inventory: 1,200	Inventory: 2,400	Inventory: 5,100
Hosts: 60	Hosts: 100	Hosts: 200	Hosts: 300
VMs: 600	VMs: 1,100	VMs: 2,200	VMs: 4,800

All performance data included in this study is from a database loaded with a year’s worth of level 4 performance statistics for the corresponding inventory.

Test Methodology

Our vCenter database performance tests simulated a live environment with regularly scheduled rollups and purges.

Initially, we populated the database with a full year's worth of level 4 performance statistics. This setting represents the worst-case scenario for vCenter database performance because level 4 collects all metrics supported by vCenter, including maximum and minimum rollup types. Level 4 performance statistics can quickly fill the vCenter database and thus adversely affect vCenter performance. We expect the level 4 performance statistics to be turned on only for limited periods of time and therefore believe that vCenter database performance should be equal to or better than the results presented in this paper.

After this initial population, we took the following steps:

- Every 30 minutes, we rolled up the five-minute statistics that were more than an hour old and purged the five-minute statistics that were more than 24 hours old.
- Every two hours, we rolled up the half-hour statistics that were more than an hour old and purged the half-hour statistics that were more than a week old.
- Every 24 hours, we rolled up the two-hour statistics that were more than an hour old and purged the two-hour statistics that were more than a month old.

We measured the response time for all of the statistics-related stored procedures while they ran as they would in a normal production environment. We were particularly interested in any delays or increases in response time that would indicate any blocking, resource contention, or deadlocks that might occur when multiple stored procedures run concurrently. We ran the benchmarks for a period of at least 24 hours, so that we could see the effect of a full cycle of rollups. We ran the benchmarks with an increasing number of entities in the vCenter Server database.

Performance Results

Our tests resulted in information about:

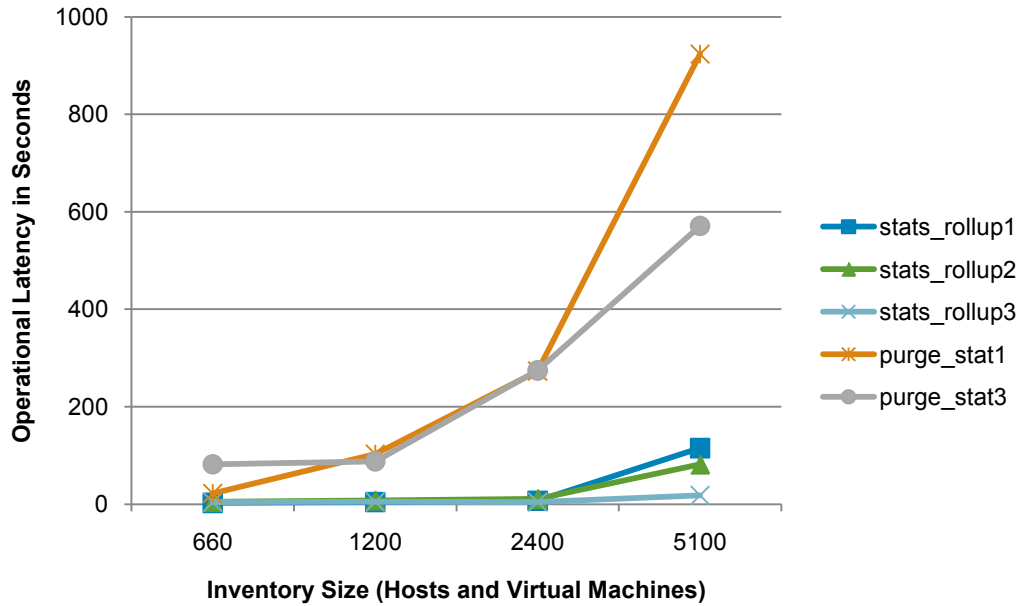
- the operational latency of statistics rollups and purge procedures
- database server memory usage in vSphere 4.0

The following sections describe our findings.

Operational Latency of Statistics Rollups and Purge Procedures

We measured the operational latency in seconds of statistics rollups and purge procedures for different inventory sizes. Our tests show that the purge procedures take an incrementally longer time as inventory size increases. Statistics rollups, however, have a negligible latency impact.

Figure 2. Performance statistics rollup and purge time comparison



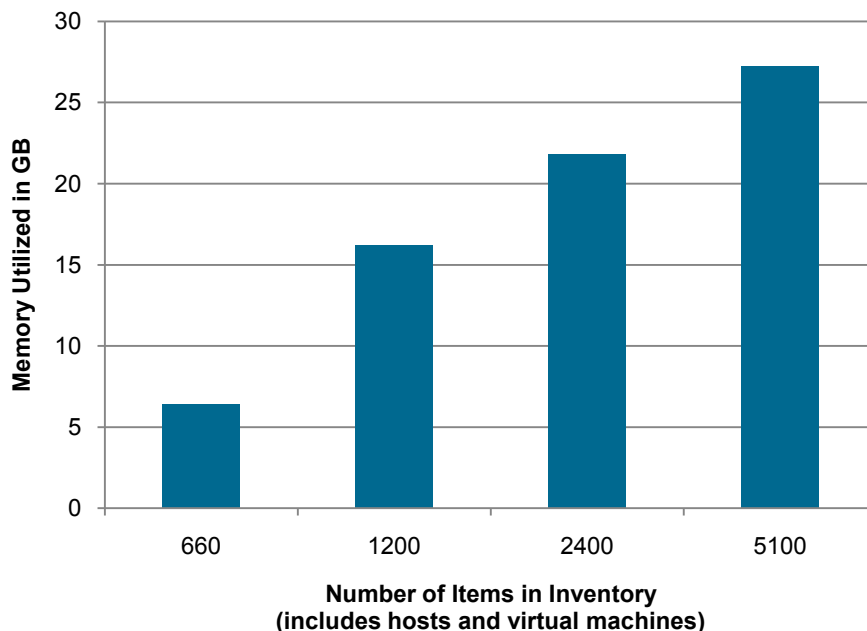
Legend:

- stats_rollup1: Runs every 30 minutes and rolls up 5 minute stats to a 30 minute stat.
- stats_rollup2: Runs every 2 hours and rolls up 30 minute stats to a 2 hour stat.
- stats_rollup3: Runs every 24 hours and rolls up 2 hour stats to a 24 hour stat.
- purge_stat1: Purges 5 minute stats that are from more than a day ago.
- purge_stat3: Purges 2 hour stats from a month ago.

Database Server Memory Usage in vCenter 4.0

Our tests show that SQL Server memory grows incrementally with inventory size. As seen in the chart below, the memory usage of the database server ranges from about 5GB with 60 hosts and 600 VMs to almost 30GB with 300 hosts and 4800 VMs.

Since 5,000 VMs and 300 hosts are the maximum supported for vCenter 4.0, the worst-case memory usage for a vCenter 4.0 system is close to 30GB.

Figure 3. Database memory for Level 4 performance statistics

Effect of Statistics Collection on Database Size

One of the most important factors in database size and performance for most vSphere 4.0 installations is the performance statistics sample size—that is, the number of performance statistics that vCenter Server gathers in each 5 minute sample period. This number is a function of the number of hosts, virtual machines, clusters, resource pools, configured devices per entity, and the configured statistics level. The database size is therefore dependent on all the inventory items and the configured statistics level.

Estimating Database Size

Using the the vCenter Server Database Sizing Calculator, you can make a reasonably accurate estimate of the database size that includes the number of performance statistics in your sample size based on the number of entities (hosts and virtual machines) in your inventory and the statistics level set in your configuration. The vCenter Server Database Sizing Calculator is an Excel spreadsheet, which you can download from http://www.vmware.com/support/vsphere4/doc/vsp_4x_db_calculator.xls.

The vCenter Client also provides a way to estimate the space needed by the vCenter Server database to store your performance statistics.

In the vSphere Client, select **Administration > vCenter Server Settings > Statistics**.

Enter the number of physical hosts and virtual machines in your inventory.

Review the value of **Estimated space required**.

Although the actual database size varies depending on a variety of factors in your environment, this figure can provide an initial estimate.

Performance Best Practices

Our testing has identified a number of best practices that can help you ensure the best performance from your vCenter database.

Separate Critical Files onto Separate Drives

When creating your SQL Server databases, make sure the database and transaction log files are placed on separate physical drives. You should also place the tempdb database on a separate physical drive if possible. This arrangement distributes the I/O to the database and dramatically improves its performance. If a third drive is not feasible, place the tempdb files on the transaction log drive.

Size tempdb

By default tempdb is 8MB and set to autogrow. Autogrow makes managing your database easier and avoids performance problems due to the data file filling up. If you have a large database and you have autogrow set to on, however, performance can suffer when the files need to grow. If this is a problem for your system, you can turn off autogrow¹, as long as you have a reliable system to monitor it and send an alert if its space is low. By setting your files large enough to begin with and monitoring data file size using an alert through SQL Agent, you can gain more control over the growth of your database and see that the performance is not affected. To size tempdb when autogrow is disabled, add 20% to the estimated space requirement (see “Estimate Database Size,” above) and use this as your starting point. For more information about tempdb performance with SQL Server 2008, see <http://msdn.microsoft.com/en-us/library/ms175527.aspx>. [7]

Use a Simple Recovery Model or Size the Transaction Log

VMware recommends using a simple recovery model for your vCenter database server. A simple model reclaims log space to keep space requirements small, essentially eliminating the need to manage the transaction log space.

The vCenter Server installer alerts the user about using a simple recovery model for the database during installation.

If you use a different recovery model setting², you should size the transaction log. The general rule of thumb for setting the transaction log size is to set it to 20-25% of the database size (see “Estimate Database Size,” above). The smaller the size of your database, the greater the size of the transaction log should be, and vice versa. Plan to back up the transaction log more often.

Enable Automatic Statistics

SQL Server contains a cost-based optimizer. To take advantage of the optimizer, enable automatic creation and updating of statistics. These statistics allow the database to optimize queries based on observed patterns of access, greatly improving performance. The overhead cost of enabling this feature is very minimal compared with the potential benefits. SQL Server enables automatic updating of statistics by default. If automatic updating is not enabled in your environment, right-click the name of the database in SQL Server Management Studio, from the context menu choose **Properties**, click the **Options** tab, then set **Auto Update Statistics** to true.

¹ In SQL Server Management Studio, right-click the vCenter database, select **Properties**, under **Select a page** on the left select **Files**, select the file you want and click “...”. Deselect the box next to **Enable Autogrowth**.

² See “Recovery Model Overview” in *SQL Server 2008 Books Online* for a description of the different recovery models.[10]

Set Performance Statistics Level

You can have different statistics levels for different statistics intervals. It is good practice to configure past-day and past-week statistics to no more than level 3, and configure past-month and past-year statistics at level 1, as shown in the table below.

Table 2. Suggested statistics levels per statistics interval

Statistics Interval	Statistics Level
Past Day	Level 2
Past Week	Level 2
Past Month	Level 1
Past Year	Level 1

You can also selectively turn off rollups for particular collection levels. This significantly reduces the size of the vCenter database and speeds up the processing of rollups. However, this approach might provide fewer performance statistics for historical data.

VMware recommends that you configure statistics at level 1. For information on configuring vCenter, see the vSphere 4.0 *Basic System Administration* guide.[1]

Set Database Retention Policy

In vCenter Server 4.0.x there is a Database Retention Policy setting which allows you to specify when vCenter Server tasks and events should be deleted. The default retention policy is to keep all tasks and events data indefinitely. To change the Database Retention Policy setting, in the vSphere Client click **Administration > vCenter Server Settings > Database Retention Policy**.

Set Fill Factor for Clustered Indexes on Performance Statistics Tables

vCenter 4.0 has four database tables for storing historical performance data: vpx_hist_stat1, vpx_hist_stat2, vpx_hist_stat3, and vpx_hist_stat4. Each of these tables has its own clustered index. You should usually set the fill factor for clustered indexes to a value that works best for your application. The fill factor option specifies how full SQL Server 2008 should make each page when it creates a new index using existing data. The fill factor affects performance because if the fill factor percentage is too large, the server must take time to split pages when they fill up. Conversely, if the percentage is too low, your database is much larger than necessary because of the unused space on each page. This increases the number of pages that need to be read during normal operations. We recommend a fill factor of 70 percent. The default is 0; values can be set from 0-100.

You can use SQL Server Management Studio to set this value for the clustered index on each of the four tables mentioned above.

To set the fill factor to 70 percent from the command line, enter the following command:

```
dbcc dbreindex( 'vpx_hist_stat1', 'pk_vpx_hist_stat1', 70)
```

Repeat this command, changing "stat1" to "stat2", "stat3", and "stat4", for each table.

Parallel Processing in SQL Server

SQL Server allows you to configure the number of threads that it can use to process a single statement. For the stored procedures that roll up and purge performance statistics to run most efficiently, we recommend the default setting of 0, which allows SQL Server to use the number of threads that it has determined to be most efficient.

To restore this value to the default from the command line, enter the following command:

```
sp_configure 'max degree of parallelism', 0
```


NOLOCK Option for Ad Hoc Queries

If you prefer to query the database with your own SQL statements, the NOLOCK option prevents queries to the performance statistics tables from being locked out when user queries are run concurrently with the rollup and purge stored procedures.

Correctly Sizing Memory Allocated to SQL Server

In general, databases run best when they are configured with enough memory to cache necessary data and reduce, if not eliminate, the need to read data from the physical disk. The best method for determining if your server's memory is configured correctly is to monitor the buffer cache hit ratio counter described in Appendix B: Monitoring SQL Server Performance with Windows Performance Management Console.

A good rule of thumb for all relational databases is to give them as much memory as necessary to allow them to cache all of their needed data in memory. Refer to your Windows documentation for the amount of memory to give to the operating system and any other applications that run on the server you use for your database. For 32-bit systems, give as much of the remaining memory as necessary to the database. The memory requirements for 64-bit systems don't change—you just need enough memory to support the system.

The commands to determine your current SQL Server memory settings and to reset the memory value are provided in Appendix A: Configuring Memory Size for 32-Bit SQL Server.

Decide on a Suitable Backup Strategy for Your Environment

If you have no backup strategy, the SQL Server transaction log continues to grow indefinitely, and in a relatively short period of time your database could become quite large. Refer to the appropriate Microsoft documentation to determine the backup strategy that best serves your needs.

References

- [1] *Basic System Administration*. VMware guide.
http://www.vmware.com/pdf/vsphere4/r40/vsp_40_admin_guide.pdf
- [2] "Fill Factor." SQL Server 2008 Books Online (November 2009).
<http://msdn.microsoft.com/en-us/library/ms177459.aspx>
- [3] "Monitoring System Resources for SQL Server." Microsoft TechNet, SQL Server TechCenter.
<http://technet.microsoft.com/en-us/library/ms191246.aspx>
- [4] *vSphere Compatibility Matrixes*. VMware documentation.
http://www.vmware.com/pdf/vsphere4/r40/vsp_compatibility_matrix.pdf
- [5] "VirtualCenter Database Maintenance: SQL Server." VMware technical note.
http://www.vmware.com/files/pdf/vc_microsoft_sql_server.pdf
- [6] "Performance and Scalability of Microsoft SQL Server on VMware vSphere 4." VMware white paper.
http://www.vmware.com/files/pdf/perf_vsphere_sql_scalability.pdf
- [7] "Optimizing tempdb Performance." SQL Server 2008 Books Online (November 2009).
<http://msdn.microsoft.com/en-us/library/ms175527.aspx>
- [8] "Predeployment I/O Best Practices: SQL Server Best Practices Article." Microsoft TechNet (June 2007).
<http://technet.microsoft.com/en-us/library/cc966412.aspx>
- [9] "Disk Partition Alignment Best Practices for SQL Server." MSDN Library (May 2009).
<http://msdn.microsoft.com/en-us/library/dd758814.aspx>

[10] "Recovery Model." SQL Server 2008 Books Online (November 2009).
<http://msdn.microsoft.com/en-us/library/ms189275.aspx>

Appendix A: Configuring Memory Size for 32-Bit SQL Server

Determining SQL Server Memory Configuration

Use the following SQL commands to determine the current setting:

```
sp_configure 'max server memory'
go
```

Setting SQL Server Memory for 32-Bit SQL Server

Use the following SQL commands to set SQL Server to use, for example, 2GB of memory:

```
sp_configure 'max server memory', 2048
go
RECONFIGURE
go
```

Configuring Large Memory Sizes for 32-Bit SQL Server

If you have AWE (Address Windowing Extensions) enabled in the host computer's Windows operating system, you can configure 32-bit SQL Server to take advantage of this extra memory, up to 6GB.

Use the following SQL commands to set 32-bit SQL Server to use 6GB of memory:

```
sp_configure 'show advanced options', 1
go
RECONFIGURE
go
sp_configure 'awe enabled', 1
go
RECONFIGURE
go
sp_configure 'max server memory', 6144
go
RECONFIGURE
go
```

Appendix B: Monitoring SQL Server Performance with Windows Performance Management Console

The Windows Performance Monitor can provide you with a wealth of information on the performance of your SQL Server database. You might find it useful to track the statistics listed in this appendix.

SQLServer:Buffer Manager

- Buffer cache hit ratio—The percentage of time that SQL Server gets the page it needs from the in-memory cache and does not have to wait to read it from disk. After the database has been running for a while, this number should always be in the high 90s. Anything less is a sign that your database is spending too much time waiting to retrieve data from the disk drives and would benefit from an increase in the amount of memory allocated to SQL Server.
- Page lookups/sec—Number of database pages requested by SQL Server. High numbers can sometimes signify that data is not properly indexed or that the query optimizer is not using the most efficient index.

Processor

- %Processor Time—Percentage of time that the CPU is busy. If consistently over 80 percent, you might be bottlenecked by CPU.

PhysicalDisk

- Avg. Disk sec/Read (latency)—The average time, in seconds, of a read of data from the disk. If the read latency increases and the throughput remains the same, it could be a case of disk saturation.
- Avg. Disk sec/Write (latency)—The average time, in seconds, of a write of data to the disk. An important counter to watch out for in your transaction log disk.
- Disk Writes/sec—Number of disk writes per second. Another counter for transaction log disk monitoring to ensure that transaction log writes are not bottlenecking your system.

For more detailed information about monitoring physical disk counters for SQL Server performance, refer to the Microsoft TechNet article “Monitoring System Resources for SQL Server.”[3]

SQLServer:Locks

- Lock Waits/sec—Number of times a lock request cannot be granted immediately. Monitor this along with the Lock Wait Time counter.
- Lock Wait Time (ms)—Number of milliseconds in the last second that a SQL Server process is blocked waiting for a lock. SQL Server uses locks to protect data in a transaction. A high number indicates high contention for the same data

SQLServer:Latches

- Latch Waits/sec—Number of times a latch request cannot be granted immediately. Monitor this along with the Latch Wait Time counter.
- Latch Wait Time (ms)—Number of milliseconds in the last second that a SQL Server process is blocked waiting for a latch. Latches are lighter weight than locks and should have much shorter, if any, wait times.

About the Author

Adarsh Jagadeeshwaran is a member of technical staff with the Performance Engineering group at VMware. He has been working on VMware vCenter Server performance. He holds a Master’s degree in Computer Science from Syracuse University.

Acknowledgements

Thanks to Ravi Soundararajan, Jeff (Xin) Liu, Shyam Venkatram, and Sabyasachi Roy for their tremendous support, Chethan Kumar, Priya Sethuraman, Rajit Kambo, Adwait Sathye and Julie Brodeur for their valuable input on this paper.

All information in this paper regarding future directions and intent are subject to change or withdrawal without notice and should not be relied on in making a purchasing decision concerning VMware products. The information in this paper is not a legal obligation for VMware to deliver any material, code, or functionality. The release and timing of VMware products remains at VMware's sole discretion.

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

Copyright © 2010 VMware, Inc. All rights reserved. This product is protected by U.S. and international copyright and intellectual property laws. VMware products are covered by one or more patents listed at <http://www.vmware.com/go/patents>. VMware, the VMware logo and design, Virtual SMP, and VMotion are registered trademarks or trademarks of VMware, Inc. in the United States and/or other jurisdictions. All **other** marks and names mentioned herein may be trademarks of their respective companies.

Item: EN-000347-00 Revision: 4/9/2010
