

VirtualCenter Database Performance for Microsoft SQL Server 2005

VirtualCenter 2.5

VMware® VirtualCenter uses a database to store metadata on the state of a VMware Infrastructure environment. Performance statistics and their associated stored procedure operations constitute the largest and the most resource-intensive component of the VirtualCenter database. Hence the performance of your VirtualCenter database depends upon the frequency at which you collect performance statistics and the level of detail of the statistics you store. VirtualCenter 2.5 features a number of enhancements that are aimed at greatly improving the performance and scalability of the performance statistics operations in the VirtualCenter database. The purpose of this study is to present the performance results of tests we conducted to validate these performance enhancements and to provide best practices information for configuring a VirtualCenter database. The study also provides information for sizing the server you use to host the VirtualCenter database based on these performance results. Although the new features in VirtualCenter 2.5 benefit users with any of the supported databases, the examples and performance data presented in this study are specific to Microsoft SQL Server and the paper assumes that you have a working knowledge of SQL Server.

This study covers the following topics:

- [“VirtualCenter 2.5 Database Overview”](#) on page 2
- [“Performance Statistics Collection in VirtualCenter”](#) on page 2
- [“Performance Statistics Database Operations and Their Effects on Performance”](#) on page 3
- [“Performance Enhancement in VirtualCenter 2.5”](#) on page 3
- [“Performance Tests”](#) on page 4
- [“Performance Results”](#) on page 5
- [“Sizing the VirtualCenter Database Server”](#) on page 7
- [“Performance Best Practices”](#) on page 8
- [“References”](#) on page 9
- [“Appendix A: Configuring Performance Statistics Levels and Rollups with VirtualCenter”](#) on page 10
- [“Appendix B: Configuring Memory Size for SQL Server”](#) on page 11
- [“Appendix C: Monitoring SQL Server Performance with Windows Performance Management Console”](#) on page 12

VirtualCenter 2.5 Database Overview

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

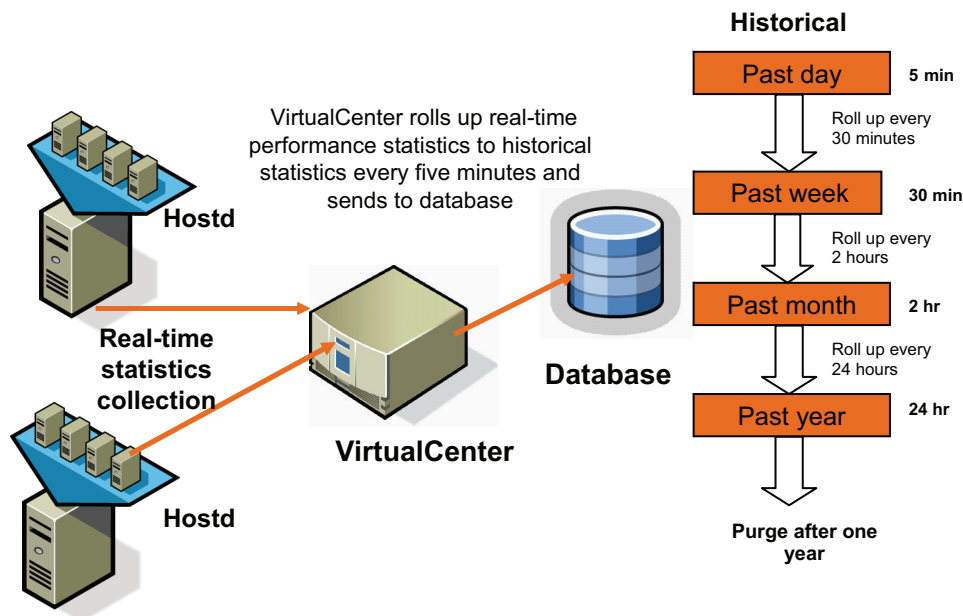
- Inventory—includes host and virtual machine configurations, resources and virtual machine inventory, user permissions, and roles. The inventory information in VirtualCenter 2.5 uses slightly more space in the database than that in VirtualCenter 2.0 because VirtualCenter 2.5 stores extra host information.
- Alarms and events—includes notifications that are triggered based on virtual machine state changes. The alarm and event data is stored indefinitely in the VirtualCenter database.
- Performance statistics—includes the performance and resource utilization statistics for datacenter elements, such as hosts, clusters, resource pools, and virtual machines. VirtualCenter provides configurable settings to control how performance statistics are collected across the VMware Infrastructure environment. Performance statistics comprise up to 90 percent of the VirtualCenter database size and therefore are the primary factor in performance and scalability of the VirtualCenter database. As a result of performance enhancements, VirtualCenter 2.5 uses less database space for the performance statistics than VirtualCenter 2.0.x does.

VirtualCenter Server supports Oracle, SQL Server, and Microsoft SQL Server 2005 Express databases. The information in this study pertains specifically to Microsoft SQL Server 2005.

Performance Statistics Collection in VirtualCenter

VirtualCenter collects real-time statistics from the host agent running on each ESX Server host. Every five minutes, VirtualCenter inserts the real-time statistics into the VirtualCenter 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 VirtualCenter database every 30 minutes, every two hours, and every 24 hours. You can 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 VirtualCenter database by:

- Changing the interval length or collection frequency
- Changing the collection level

- Enabling or disabling a collection interval

Because the VirtualCenter database size is the single most important factor in determining the performance and scalability of VirtualCenter, you should configure these settings carefully. For further details, see “Virtual Center Monitoring and Performance Statistics.” For a link to that paper see “References” on page 9.

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**—VirtualCenter inserts performance statistics into the database every five 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 percent is typical for the VPX_HIST_STAT tables.

Performance Enhancement in VirtualCenter 2.5

VirtualCenter 2.5 features extensive changes to the performance statistics database tables, stored procedures, and indexes. These changes provide:

- Increased performance and scalability of performance statistics operations.
- A restructured VPX_HIST_STAT table, which is now four separate tables, each with a clustered index that both reduces database size, resulting in lower memory and smaller on-disk footprint, and avoids the performance penalty of fragmented indexes.
- Significantly reduced space requirements for tempdb, which is no longer a critical factor in database performance.

This study presents the performance results for the first two improvements listed above.

Performance Tests

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

Test Environment

Hardware

- 2 Intel Xeon CPU X5355
- 8GB SQL Server configured memory

Software

- Operating system: Microsoft Windows Server 2003 Enterprise Edition SP 1
- Microsoft SQL Server 2005 SP 2 32-bit
- AWE enabled

NOTE VirtualCenter 2.0.1 supports only the 32-bit version of SQL Server 2005, hence we used the 32-bit version for our performance testing. VirtualCenter 2.0.2 and 2.5 also support the 64-bit version of SQL Server 2005. When you use the 64-bit version, you do not need to enable AWE when using more than 4GB of memory. See the *ESX Server, VirtualCenter, and VI Client Compatibility Guide* for more details. For a link to the guide, see [“References”](#) on page 9.

Benchmark Setup

- 10 virtual machines per host
- All performance data shown is from a database loaded with a year’s worth of level 4 performance statistics

Test Methodology

Our VirtualCenter database performance tests used a live environment with inserts at five-minute intervals plus 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 VirtualCenter database performance because level 4 collects all metrics supported by VirtualCenter, including maximum and minimum rollup types. Level 4 performance statistics can quickly fill the VirtualCenter database and thus adversely affect VirtualCenter performance. We expect the level 4 performance statistics to be turned on only for limited periods of time and therefore believe that VirtualCenter 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 five minutes, we inserted a new sample period and corresponding statistics (at level 4, this means about 100 performance statistics for each host and each virtual machine).
- 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.

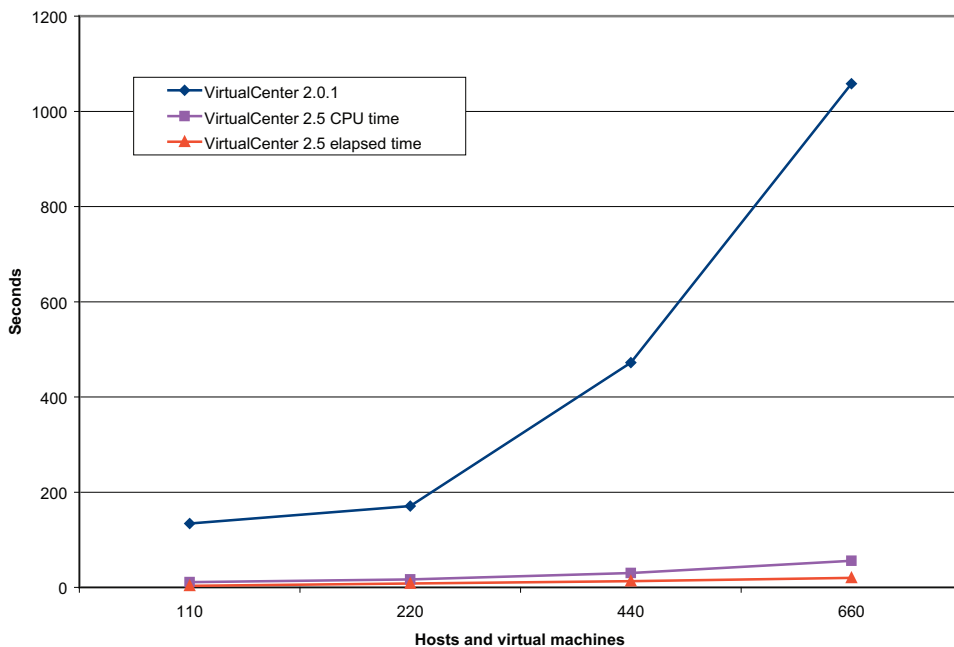
We ran the benchmarks with an increasing number of entities in the VirtualCenter database. We assumed that 10 virtual machines were hosted per ESX Server host. For example, the test with 110 entities consisted of 10 ESX Server hosts and a total of 100 virtual machines.

Performance Results

As Figure 2 shows, VirtualCenter 2.5 provides greatly improved performance compared to VirtualCenter 2.0.x.

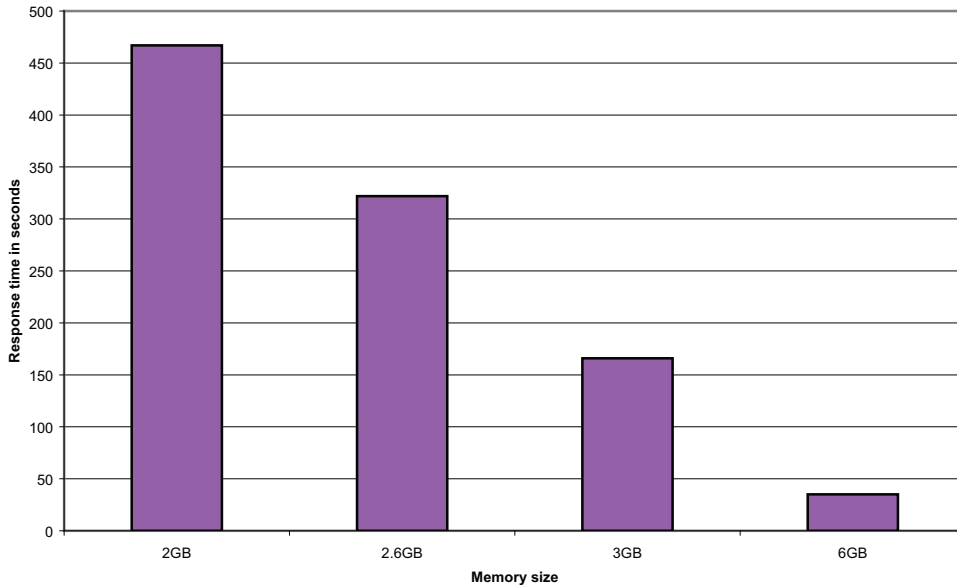
- Rollup times are significantly shorter.
- Rollup procedures can now use the parallel processing feature in SQL Server so elapsed time is often much shorter than CPU time.
- Displaying VirtualCenter 2.5 graphs for historical performance statistics is no longer impeded by the rollup procedures.
- Rollup times scale linearly as you increase inventory or statistics levels.

Figure 2. Performance Statistics Rollup Time Comparison



NOTE In VirtualCenter 2.0.x CPU time and elapsed time are the same.

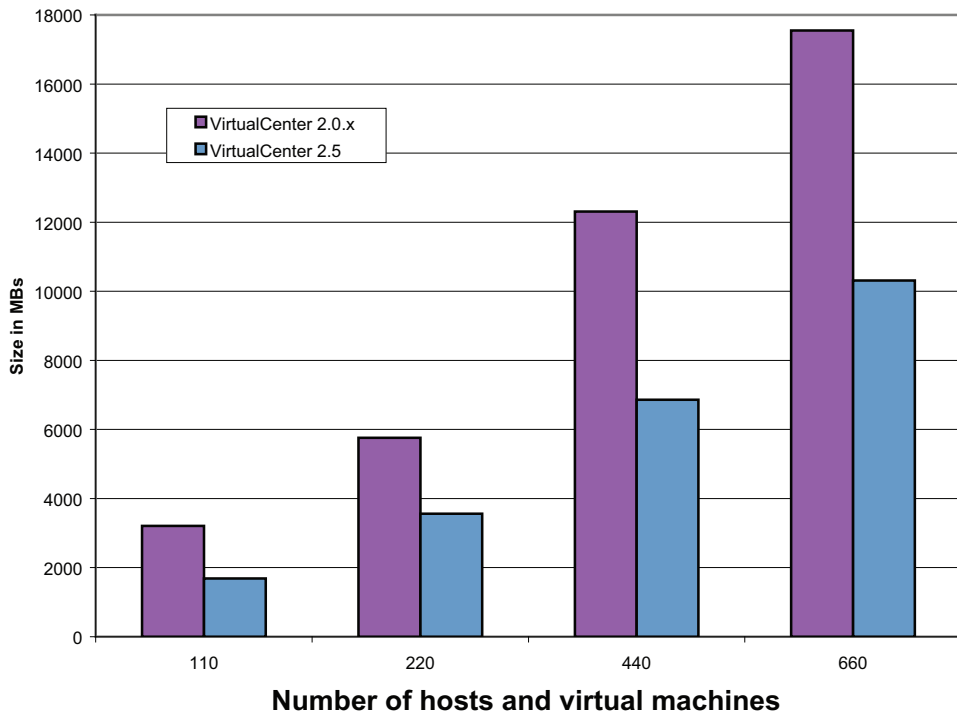
The size of a database's in-memory buffer cache plays a crucial role in the performance of any database. The operation most affected by memory size is the statistics rollups. Figure 3 shows that the average response time for the 30-minute rollup with 2GB of memory was more than 450 seconds, as compared to about 40 seconds for the rollup with 6GB of memory, a 10-fold improvement in the response time when the database memory was increased only three times. With smaller memory configurations, the data being rolled up is replaced in the database's buffer cache with more recent data and needs to be fetched from disk. Optimally, you should have a large enough cache so the whole database fits into memory. Another option is to reduce the size of the data being stored by controlling the performance statistics collection interval and level.

Figure 3. Impact of Memory Size on Rollup Time

Database Size Improvement in VirtualCenter 2.5

The single most important factor in database size and performance for most VirtualCenter 2.5 installations is the performance statistics sample size—that is, the number of performance statistics that VirtualCenter gathers in each five-minute sample period. This number is a function of the number of hosts, the number of virtual machines, and the configured statistics level.

VirtualCenter 2.5 uses less database space for the performance statistics than VirtualCenter 2.0.x did. Figure 4 compares the database sizes of the two VirtualCenter versions when they have a year's worth of Level 4 performance statistics.

Figure 4. Database Size for Level 4 Performance Statistics

Sizing the VirtualCenter Database Server

Figuring out the performance statistics sample size is the key to sizing the VirtualCenter database server successfully. There are two ways to determine the sample size for your VirtualCenter configuration.

- Estimate sample size based on number of entities and statistics level
- Check actual sample size by querying VirtualCenter database

Estimate Based on Number of Entities and Statistics Level

You can make a reasonably accurate estimate of 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.

Using Table 1, You can calculate a close approximation of this value with the following formula:

sample size = number of entities (hosts + virtual machines) * statistics per statistics level

Table 1. Performance Statistics per Level

Statistics Level	1	2	3	4
Statistics per entity	10	25	65	100

NOTE Statistics for levels 3 and 4 vary depending on the number of configured devices.

Example

100 hosts with 500 total virtual machines at statistics level 2 = 600 * 25 = 15,000 statistics per sample period

Determining Sample Size by Querying VirtualCenter Database

If you already have VirtualCenter 2.5 running and your inventory in place, you can get the exact performance statistics sample size by directly querying SQL Server using the commands shown below.

SQL for VirtualCenter 2.5

```
select count(*) from vpx_hist_stat1
where time_id in (select (MAX(time_id) - 1) from vpx_sample_time1)
go
```

SQL for VirtualCenter 2.0

```
select count(*) from vpx_hist_stat where sample_id in (select (max(id) - 1) from vpx_sample)
```

Hardware Guidelines for Sample Sizes

In order to achieve optimal database performance for your VirtualCenter configuration, you should size your hardware appropriately. Table 2 provides recommendations you can use to determine the hardware resources you need to achieve optimal performance from your VirtualCenter database

Table 2. Hardware Sizing Guidelines

Sample Size	CPU Cores	Memory Allocated to SQL Server	Log Device Disk I/O Writes/Second
< 40K	1	1GB	< 130
40K–80K	2	2GB	120–200
80K–120K	4	4GB	200–300
120K–160K	4	6GB	400–500
160K–200K	6	8GB	500–600
> 200K	8	12GB	> 650

Performance Best Practices

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

Estimate Database Size

Use the **Estimate database space** tool in the Statistics Collection Settings panel to generate an estimate of the space required to store your performance statistics in the VirtualCenter database. Although the actual database size varies depending on a variety of factors in your environment, this figure can provide an initial estimate.

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.

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.

Set Performance Statistics Level

VirtualCenter 2.5 provides more granular control for performance statistics. You can have different statistics levels for different statistics intervals. It is good practice to configure past-day and past-week statistics at level 3, and configure past-month and past-year statistics at level 1. You can also selectively turn off rollups for particular collection levels. This significantly reduces the size of the VirtualCenter database and speeds up the processing of rollups. However, this approach might provide fewer performance statistics for historical data.

Set Fill Factor for Clustered Indexes on Performance Statistics Tables

VirtualCenter 2.5 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 2005 should make each page when it creates a new index using existing data. The fill factor affects performance because if the fill 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.

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)
```

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 C: Monitoring SQL Server Performance with Windows Performance Management Console" on page 12. The commands to determine your current SQL Server memory settings and to reset the memory value are provided in "Appendix B: Configuring Memory Size for SQL Server" on page 11.

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. See "Virtual Center Database Maintenance: SQL Server" (for a link, see "References" on page 9) or the appropriate Microsoft documentation to determine the backup strategy that best serves your needs.

References

- "Monitoring System Resources for SQL Server"
<http://technet.microsoft.com/en-us/library/ms191246.aspx>
- *Basic System Administration*
http://www.vmware.com/pdf/vi3_35/esx_3/r35/vi3_35_25_admin_guide.pdf
- *ESX Server, VirtualCenter, and VI Client Compatibility Guide*
http://www.vmware.com/pdf/vi3_35/esx_3/r35/vi3_35_25_compat_matrix.pdf
- "Virtual Center Database Maintenance: SQL Server"
http://www.vmware.com/files/pdf/vc_microsoft_sql_server.pdf

Appendix A: Configuring Performance Statistics Levels and Rollups with VirtualCenter

The VirtualCenter 2.5 Management Service UI enables you to configure statistics collection levels in the following ways:

- Configure collection levels separately for each interval
- Turn off rollups for particular collection levels
- Configure existing levels in certain ways
 - Change the collection frequency for the five-minute statistics to 1, 2, 3, or 5 minutes
 - Keep the five-minute statistics for up to five days rather than the default of one day
 - Keep the one-day statistics for up to five years rather than the default of one year

VirtualCenter 2.5 allows you to configure performance statistics levels for various time periods. For example, if you want to have detailed statistics for the past day and the past week, but anything older is not as important, you can configure past day statistics to be collected at level 3, past week statistics at level 3, but past month and past year statistics at level 1. This significantly reduces the size of your VirtualCenter database.

VirtualCenter 2.5 enables you to selectively turn off rollups by unchecking specific collection intervals. For example, if you have no interest in performance statistics that are more than a week old, you can disable rollup at the past-month level, which will also disable all rollups at longer intervals (past year in this case).

For additional information on configuring VirtualCenter, see the VMware Infrastructure *Basic System Administration* guide. For a link to the guide, see “[References](#)” on page 9.

Database Size Estimator

Open the VirtualCenter Management Server Configuration window and choose Statistics to display the Statistics Collection Settings panel. After making the desired settings and entering the number of ESX Server hosts and virtual machines, click **Estimate database space** to generate an estimate of the space required to store your performance statistics in the VirtualCenter database. The figure is approximate. Actual size varies depending on a variety of factors in your environment.

Appendix B: Configuring Memory Size for SQL Server

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. Give as much of the remaining memory as necessary to the database.

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 to 2GB

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

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

Configuring Large Memory Sizes for 32-Bit SQL Server 2005

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.

Configuring 32-Bit SQL Server to Use 6GB of Memory

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 C: Monitoring SQL Server Performance with Windows Performance Management Console

The Windows Performance Monitor can provide the 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

- Current Disk Queue Length—The number of outstanding disk I/Os at the time the data is collected. If this number is consistently greater than two, your database might be bottlenecked by disk I/O.
- Disk Writes/sec—Number of disk writes per second. Important counter for transaction log disk monitoring to ensure that transaction log writes are not bottlenecking your system.

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.

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

Copyright © 2008 VMware, Inc. All rights reserved. Protected by one or more of U.S. Patent Nos. 6,397,242, 6,496,847, 6,704,925, 6,711,672, 6,725,289, 6,735,601, 6,785,886, 6,789,156, 6,795,966, 6,880,022, 6,944,699, 6,961,806, 6,961,941, 7,069,413, 7,082,598, 7,089,377, 7,111,086, 7,111,145, 7,117,481, 7,149,843, 7,155,558, 7,222,221, 7,260,815, 7,260,820, 7,269,683, 7,275,136, 7,277,998, 7,277,999, 7,278,030, 7,281,102, and 7,290,253; patents pending. VMware, the VMware "boxes" logo and design, Virtual SMP and VMotion are registered trademarks or trademarks of VMware, Inc. in the United States and/or other jurisdictions. Microsoft, Windows and Windows NT are registered trademarks of Microsoft Corporation. Linux is a registered trademark of Linus Torvalds. All other marks and names mentioned herein may be trademarks of their respective companies.
Revision 20080411 Item: PS-054-PRD-01-01
