



VMware[®] vCenter Server[™] 5.1 Database Performance Improvements and Best Practices for Large-Scale Environments

Performance Study

TECHNICAL WHITE PAPER

Table of Contents

Executive Summary	3
Introduction	3
Overview of Statistics Subsystem in vCenter.....	4
Database Performance Improvements	6
Removed Staging Tables.....	6
Partitioned Statistics Tables.....	7
Redesigned Stored Procedures.....	8
Database Best Practices for Large-Scale Environments.....	8
How to Lay Out Disks for Both Oracle and SQL Server	8
How to Update Index Statistics for Highly Volatile Tables	9
How to Separate Tables and Indexes for Improved Performance.....	10
How to Take Advantage of Features in Enterprise Editions of SQL Server	10
Conclusion	11
Appendix: Test Bed Configuration	11
References.....	12

Executive Summary

VMware® vCenter Server™ 5.1 introduces some significant improvements to the statistics subsystem. Statistics data represents the largest storage impact in a vCenter Server database and therefore vCenter Server must handle this data in an effective way so that it does not hamper VMware vSphere® performance. To help meet this need, vCenter Server 5.1 reduces the resource overhead of the database through enhancements in stored procedures in the database, specifically, the rollup and TopN procedures. This paper describes these changes and provides the following best practices for configuring the database to best take advantage of the improvements:

- How to lay out disks for both Oracle and SQL Server
- How to update index statistics for highly volatile tables
- How to separate tables and indexes for improved performance
- How to take advantage of features in enterprise editions of SQL Server
- How to tune certain parameters for Oracle and SQL Server, for example, the parallelism threshold for SQL Server

The improvements are especially important for vCenter Server 5.1 deployments running at-scale inventory; that is, at or near the maximum number of inventory (virtual machines, hosts, datastores, clusters) that a vCenter Server can manage.

Introduction

VMware vCenter Server provides a centralized way to control and monitor the virtual infrastructure. It persists certain important information into a relational database. This data falls into four categories: 1) inventory and configuration data, 2) task and event data, 3) alarm data, and 4) statistics data.

The database is therefore a critical component of vCenter Server performance. Because the statistics data consumes a large fraction of the database, proper functioning of statistics is an important consideration for the overall database performance. Thus, statistics collection and processing are key components for vCenter Server performance.

This paper focuses on the statistics subsystem of the vCenter Server. The first part of the paper describes the statistics subsystem and a number of significant improvements to the statistics subsystem in vCenter Server 5.1. The second half of the paper describes more general best practices for overall database performance.

Overview of Statistics Subsystem in vCenter

Figure 1 shows an overview of the statistics subsystem in vCenter Server.

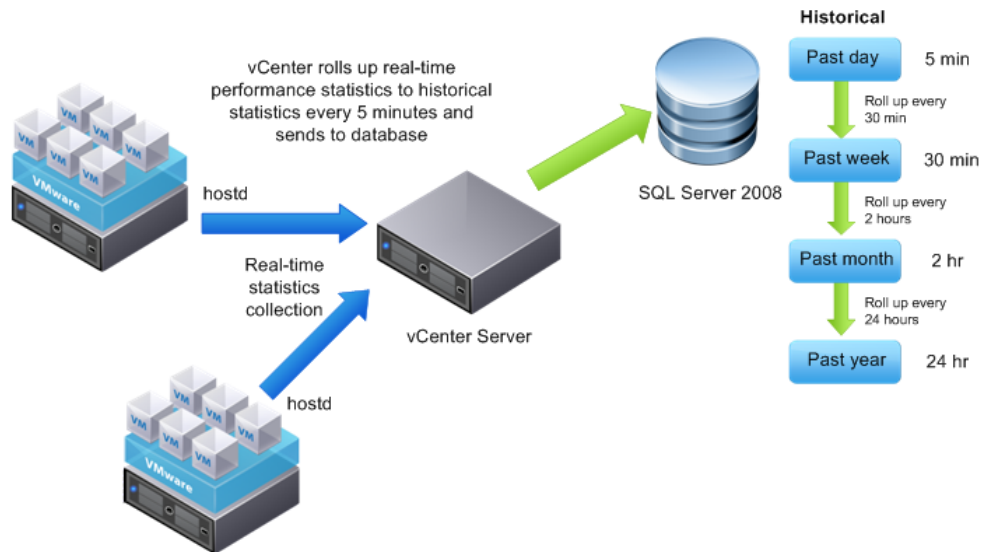


Figure 1. Statistics subsystem in vCenter Server

Each ESXi host is connected to a vCenter Server, and vCenter Server is connected to a relational database. vCenter Server collects statistics from each ESXi host periodically and persists this data to the relational database. The database in turn executes a number of stored procedures that summarize this data at various intervals.

Each ESXi host collects statistics at a 20-second granularity. In vCenter, these are called *real-time statistics*. You can view real-time statistics through vSphere Client by selecting the Advanced button on the Performance tab. The client always receives real-time statistics directly from the ESXi host. This ensures timeliness of the data and puts no stress on the database.

Periodically, these 20-second statistics are aggregated into 5-minute statistics. vCenter Server stores these 5-minute historical statistics in the Past Day Table. The procedure of converting a set of fifteen 20-second real-time statistics into a single 5-minute historical statistic is called a *rollup*.

There are several granularities of statistics that are stored in the vCenter Server database. In a similar manner to rolling up the 20-second statistics into the 5-minute statistics, in the background, the database executes stored procedures periodically to roll up the statistics into larger granularities:

- The Past Day statistics rollup procedure runs every 30 minutes to aggregate the 5-minute historical statistics into 30-minute historical statistics.
- The Past Week statistics rollup procedure runs every 2 hours to aggregate the 30-minute historical statistics into 2-hour historical statistics.
- The Past Month statistics rollup procedure runs every 1 day to aggregate the 2-hour historical statistics into 1-day historical statistics.

All statistics stored in the database are called *historical statistics*. You can view the historical statistics through the vSphere Client by selecting the Advanced button on the Performance tab, and then changing the chart options. The client receives historical statistics data directly from the database.

There are two key settings for the statistics subsystem in vCenter Server: the retention interval and the statistics collection level.

- **Retention Interval**

This specifies how long statistics are stored in the database. When the data is older than the retention interval, it is considered to be expired and deleted from the database.

- Past Day (5-minute) statistics are stored for 1-5 days.
- Past Week (30-minute) statistics are stored for 1 week.
- Past Month (2-hour) statistics are stored for 1 month.
- Past Year (1-day) statistics are stored for 1-5 years.

- **Statistics Collection Level**

In order to reduce the volume of the statistics data going into the database, and also to allow control of this volume, vCenter Server has statistics collection levels, which vary from 1 to 4. In general, the higher the level, the more detailed the statistics and therefore the more volume of the statistics that must be stored in the database. [Table 1](#) shows more detail for each of these levels.

- Level 1 is the least-detailed statistics level and only includes the most critical statistics, such as aggregate CPU, memory, and network usage.
- Level 2 introduces a number of additional statistics.
- Level 3 incorporates per instance statistics, for example CPU usage of a host on a per-CPU basis.
- Level 4 is the most detailed and is inclusive of all the other levels.

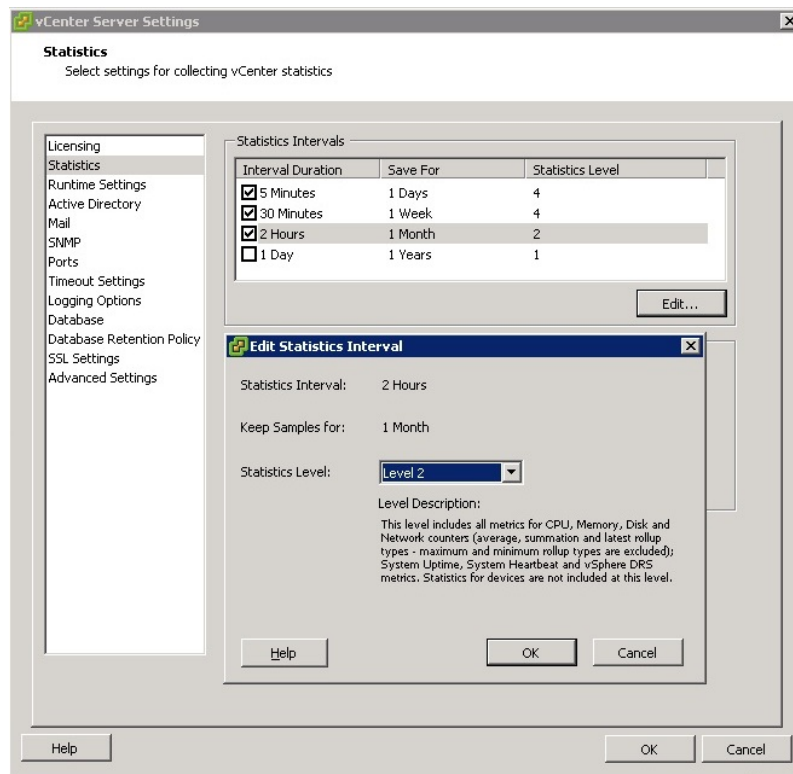


Figure 2. Dialog box used to specify different retention intervals for each statistics collection level

STATISTICS COLLECTION LEVEL	DESCRIPTION
Level 1	Includes basic metrics: Average Usage for CPU, Memory, Disk, and Network; System Uptime, System Heartbeat, and DRS metrics. Does not include statistics for devices.
Level 2	Includes all metrics for CPU, Memory, Disk, and Network counters (average, summation and latest rollup types; does not include maximum and minimum rollup types); System Uptime, System Heartbeat, and DRS metrics. Does not include statistics for devices.
Level 3	Includes all metrics (including device metrics) for all counter groups (average, summation and latest rollup types; does not include maximum and minimum rollup types).
Level 4	Includes all metrics supported by vCenter Server, including maximum and minimum rollup types.

Table 1. Description of statistics collection levels

Database Performance Improvements

This section describes a number of significant improvements to the statistics subsystem in vCenter Server 5.1. vCenter Server collects and maintains a very large amount of data, making statistics collection and persistence of this data crucial to database performance. In the vSphere 5.1 release, there are two main improvements:

- A reduced resource overhead of the database through improvements in stored procedures in the database; specifically, the rollup and TopN procedures
- More efficient support for higher statistics levels than before

These improvements are realized through three optimizations:

- Removed staging tables
- Partitioned statistics tables
- Redesigned stored procedures

Removed Staging Tables

vSphere 4.1 and vSphere 5.0 use staging tables, which accommodate the bursty behavior of the statistics collection on large-scale infrastructures. There are three of these staging tables. One staging table is used for vCenter Server to insert all the 5-minute statistics. After a fixed time interval, it switches over to the next staging table. Concurrently, the staging table that was full is parsed and all of the 5-minute statistics are inserted into the Past Day table. The third staging table is used as an extra buffer to make the transition smooth between staging tables. However, a more scalable solution is needed to meet the demands of the larger inventories supported in vSphere 5.1.

In order to address this issue, vSphere 5.1 removes these staging tables and instead partitions the statistics tables. With this change, vCenter Server now inserts 5-minute statistics directly into the Past Day Table. This change improves the overall statistics collection system significantly.

The removal of staging tables also provides for more robust statistics data retention for large-scale environments with certain network and storage configurations which experienced some loss of statistics data. See the following VMware knowledge base articles for more information: [KB 2011523](#), [KB 1003878](#).

[Table 2](#) shows the number of statistics that can be inserted per hour for different statistics levels. For example, 80 million statistics per hour are collected at level 4 for the at-scale inventory described in [Appendix: Test-Bed](#)

Configuration for 1,000 hosts (each with 32 CPUs, 2,000 datastores, and 4 NICs) and 10,000 powered-on VMs (each with 1 vCPU, 1 disk, and 1 NIC). You will observe a different number of statistics in your setup based on the configuration of your hosts and VMs. The table also shows I/O activity (KBps) for each level of statistics.

STATISTICS COLLECTION LEVEL	# OF STATISTICS / HOUR	# OF STATISTICS / 5-MIN	I/O ACTIVITY
Level 1	2 M	0.17 M	7,000 KBps
Level 2	15 M	1.25 M	14,000 KBps
Level 3	68 M	5.67 M	40,000 KBps
Level 4	80 M	6.67 M	50,000 KBps

Table 2. The number of statistics that vCenter Server collects and pushes into the database for different levels for the at-scale inventory in [Appendix: Test-Bed Configuration](#).

The removal of staging tables allows for a redesigned statistics insertion logic, which reduces the resource overhead of the database and increases the number of statistics that are persisted into the database at once, thereby increasing the scalability of the vCenter Server.

Partitioned Statistics Tables

There were three sources of I/O to the statistics tables in vCenter Server—inserting statistics, rolling up statistics between different intervals, and deleting statistics when they expire. This I/O resulted in a contention for the statistics tables, and this contention could result in highly variable and long latencies for these operations. Originally, there was a single table for each of Past Day, Past Week, Past Month and Past Year statistics, and this single table could grow very large with at-scale inventories. vSphere 5.1 includes redesigned and partitioned statistics tables, which reduce contention and improve performance.

TABLE	TIME INTERVAL FOR EACH SUB-TABLE	# OF TIME IDS IN EACH SUB-TABLE	TIME DURATION FOR STATISTICS
Past Day	Half an hour	6	5-min stats
Past Week	2 hours	4	half-hour stats
Past Month	1 day	12	2-hour stats
Past Year	10 days	10	1-day stats

Table 3. Statistics tables are partitioned into smaller sub-tables. Each holds statistics for a short time interval. For example, the Past Day statistics table is now partitioned into sub-tables, where each sub-table holds only half an hour's worth of statistics.

Due to these changes in vSphere 5.1:

- Insertions are improved dramatically.
- Rollup performance is improved significantly. Now, the rollup process is finely tuned such that the rollup procedure's performance scales with the statistics level, and it is also extremely robust, as [Table 4](#) shows that rollup takes minutes.
- Purge performance is dramatically improved, virtually eliminating all I/O to the disk. Deleting the expired statistics is now a simple truncation of a sub-table when its data is expired. This reduced the purge time to sub-seconds.
- Higher statistics levels are more efficiently supported than before.

Type of Rollup	TIME (MINUTES:SECONDS)			
	Level 1	Level 2	Level 3	Level 4
Past Day Rollup	00:30	01:20	07:30	10:50
Past Week Rollup	00:50	02:20	09:30	21:30
Past Year Rollup	03:40	09:10	23:30	24:20

Table 4. Average time to compute rollup procedures for different statistics collection levels on SQL Server.

Redesigned Stored Procedures

In addition to the changes previously described, vSphere 5.1 includes redesigned stored procedures that are more efficient than in previous releases. For example, in the datacenter and cluster charts in the client, it is possible to see “top 10” virtual machines organized by CPU usage. This chart is computed by TopN queries that do some math on the statistics to determine the “top N” virtual machines by CPU usage, memory, etc. and stored in the TopN_Day table. Periodically, these daily TopN statistics are rolled up into per week, per month, and per year tables.

These TopN procedures have been rewritten to be more efficient. In the past, they might have taken tens of minutes, but each one of these TopN procedures now takes less than a minute to complete.

The result of these changes is improved UI performance of loading pages as well as reduced I/O on the database.

Database Best Practices for Large-Scale Environments

This section provides best practices for configuring the database to best take advantage of the improvements in vSphere 5.1, including: how to lay out disks for both Oracle and SQL Server, how and when to recompute statistics on various tables, how to separate tables and indexes for improved performance, and how to tune certain parameters for Oracle and SQL Server; for example, the parallelism threshold for SQL Server.

For general vCenter Server best practices, refer to the [vSphere 5.1 Performance Best Practices Guide](#).

How to Lay Out Disks for Both Oracle and SQL Server

vCenter Server operations could cause a lot of I/O on the database server. It is recommended to spread the LUNs/disks evenly and lay the database files accordingly. Following are the guidelines.

Oracle

It is recommended to have 7 disks:

- /u01 - system01.dbf, undotbs01.dbf
- /u02 - sysaux01, temp01.dbf
- /u03 - vpxdata01.dbf
- /u04 - vpxindx01.dbf
- /oralog - redo01a.log, redo02a.log, redo03a.log
- /oralog_mirror - redo01b.log, redo02b.log, redo03b.log
- /oraarch - archive destination.

SQL Server

It is recommended to have 5 disks:

- mssql01 - master and msdb databases(.mdf, .ldf)
- mssql02 - tempdb(.mdf, .ldf), also set the initial size to 10GB
- mssql03 - VCDB(.mdf, .ldf)
- mssql04 - VCDB Backup location

How to Update Index Statistics for Highly Volatile Tables

SQL Server

The cost-based optimizer (CBO) in Microsoft SQL Server uses statistics about tables and indexes to compute the most efficient access plan. In SQL Server, index statistics are updated automatically by a database option `AUTO_UPDATE_STATISTICS`. This setting is true by default. SQL Server updates the out-of-date statistics based on the number of inserts, updates, and deletes that have occurred since the statistics were last taken, and then recreates the statistics based on a threshold. The threshold is relative to the number of records in the table. With large tables (those with a million or billion rows), SQL Server requires that a few thousand or millions of rows need to be inserted, updated, or deleted before the statistics are automatically updated. This could affect vCenter Server operations.

A few tables in the vCenter Server schema change data at a very rapid rate based on certain vCenter Server operations, and index statistics on these tables quickly become out of date. This could cause database performance to degrade.

For example `VPX_PROPERTY_BULLETIN`, `VPX_ALARM`, `VPX_EVENT`, and `VPX_EVENT_ARG` are a few of the most volatile tables in the vCenter Server database schema.

Because of the size of the tables, SQL Server might have trouble automatically keeping the statistics up to date. To avoid this issue, manually update the index statistics on highly volatile tables to use the optimal execution plan.

To update statistics for the database, use:

```
sp_updatestats VCDB;
```

To update statistics on a table, use:

```
UPDATE STATISTICS <table_name>
```

For example:

```
UPDATE STATISTICS VPX_PROPERTY_BULLETIN;
```

Oracle

The cost-based optimizer (CBO) determines the best available plan for data access, but it depends on the statistics being up to date. Stale statistics could cause an adverse effect on the database response. The default setting for the Oracle (10g, 11g) database is to gather statistics automatically. Automatic optimizer statistics collection should be sufficient for most database objects being modified at a moderate speed, however, in few cases the statistics collection is not adequate because the statistics collection job runs during maintenance windows, and also data on very large tables change very rapidly. The statistics on these tables become stale very quickly.

The vCenter Server database content changes very rapidly on certain vCenter Server operations. It is recommended to gather statistics at regular intervals to ensure that the statistics accurately represent characteristics of database objects.

The Oracle's DBMS_STATS package can be used to gather statistics on tables, indexes, and individual columns of a table.

While the statistics are being updated on a table or index, Oracle invalidates any currently parsed SQL statements that are accessing the table or index, but next time when a similar SQL statement executes, the SQL statement is re-parsed and the optimizer automatically chooses the new execution plan based on the new statistics that are available.

To update table or index statistics for Oracle 10g and later, use the Oracle package DBMS_STATS.

To gather statistics at the schema level, use the GATHER_SCHEMA_STATS procedure. For example:

```
exec.dbms_stats.gather_schema_stats
    (ownname = 'VCDB',
    estimate_percent = 20,
    method_opt = 'for all columns size auto',
    options = 'gather',
    cascade = true);
```

How to Separate Tables and Indexes for Improved Performance

SQL Server

For very large databases and highly transactional databases, you can try moving non-cluster indexes and tempdb onto their own file groups. This option was not tested in our labs but should work. Since this is an experimental procedure, and because it changes the physical database files, make sure you back up your database before attempting this procedure.

Oracle

In addition to following the recommended disk layout previously described, it is recommended to separate indexes from /u03 (data file). Testing with at-scale inventory has shown a significant improvement in database response time.

How to Take Advantage of Features in Enterprise Editions of SQL Server

SQL server uses parallel query processing in order to take advantage of machines with multiple CPUs. This method improves query and index operations by running several threads in parallel across the CPUs. Parallel execution plans can use more than one thread, whereas serial execution plans can run only one thread.

In SQL server, using **max degree of parallelism** limits the number of processors to use in parallel execution. The **max degree of parallelism** option determines the computing and thread resources used for parallel query processing.

The **cost threshold for parallelism** option specifies the threshold at which SQL Server creates and runs parallel plans for queries. SQL Server creates and runs a parallel plan for a query only when the estimated cost to run a serial plan for the same query is higher than the value set in **cost threshold for parallelism**.

SQL Server Enterprise Edition features

1. For a system with four or more processors, set the **max degree of parallelism** as follows:

```
sp_configure 'max degree of parallelism', (N/2-1);
```

N is the number of processors, and this number must be greater than or equal to 4.

2. Set the **cost threshold for parallelism** as follows:

```
sp_configure 'cost threshold for parallelism', 15;
```

15 is the recommended value; it can be higher but not more than 25.

Conclusion

This paper describes several optimizations to the vCenter Server database that involve redesigning the tables and revising the stored procedures for the statistics subsystem. With the help of these changes, vCenter Server 5.1 now:

- Provides better scalability and support for larger environments
- Provides reduced resource overheads on the database
- Is more efficient in how it does statistics collection and processing
- Is able to handle higher statistics collection levels for larger environments

And finally, this paper includes best practices for configuring the database to best take advantage of these improvements.

Appendix: Test Bed Configuration

The lab environment used the following at-scale inventory, where the vCenter Server inventory was at or near the configuration maximums for most items on the list:

- 1,000 hosts
- 10,000 powered on VMs
- 15,000 registered VMs
- 32 clusters
- 2,000 datastores

Virtual machine running vCenter Server:

- 16 vCPUs
- 52GB memory

Virtual machine running database (Microsoft SQL Server 2008):

- 16 vCPUs
- 26 GB memory
- Database files: RAID-5 storage backed up by 10 disks. Each disk is 2Gbps, 5200rpm, 133.68GB
- Log files and tempdb: 10 disks RAID-5 storage. Each disk is 2Gbps, 5200rpm, 133.68GB

References

- [1] A. Jagadeeshwaran, "VMware vCenter 4.0 Database Performance for Microsoft SQL Server 2008," 2010.
http://www.vmware.com/pdf/vsp_4_vcdb_sql2008.pdf.
- [2] "VirtualCenter Monitoring and Performance Statistics."
http://www.vmware.com/pdf/vi3_monitoring_statistics_note.pdf.
- [3] R. Vijayaraghavan, B. Parimi and J. Cook, "StatsFeeder: An Extensible Statistics Collection Framework for Virtualized Environments," *VMware Technical Journal*, vol. 1, no. 1, 2012.

About the Authors

Alper T. Mizrak is a Staff Engineer with the Performance Engineering group at VMware. He joined VMware, Research & Development in 2007 and he has been working on the performance and scalability of VMware products. Dr. Mizrak received his Ph.D. in Computer Science in December 2007 from the University of California, San Diego. During his graduate studies, he conducted research in the area of network security and distributed systems. His graduate research resulted in receipt of the William C. Carter Award—an award granted by the IEEE Technical Committee on Fault Tolerant Computing and the IFIP WG-10.4 on Dependable Computing and Fault Tolerance - to recognize an individual who has made a significant contribution to the field of dependable computing through his or her graduate dissertation research; a book, "Secure Networking: Detecting Malicious Routers," published in 2008; 8 academic papers published in peer-reviewed journals, conferences and workshops; and 3 technical reports.

Raju Angani is a Staff Database Engineer with the vCenter Database Platform Engineering group at VMware. In this role, he has been focusing on vCenter Database design, optimization and performance tuning. His past papers are on "How to deploy Oracle RAC on ESX machines ESX 3.0/4.0," and "How to deploy Microsoft SQL Server Clusters on ESX 3.0/4.0."

Acknowledgements

This white paper is the result of a collaborative effort among various teams within VMware, including the core vCenter Server team, the vCenter Server Platform Database team, as well as the vCenter Server Performance team. The authors would like to thank Ravi Soundararajan, Kinshuk Govil, Leonid Livshin, Reza Taheri, Chirag Bhatt, Adarsh Jagadeeshwaran, Sundeep Tiyyagura and Bobbie Morrison for reviewing the document and providing insightful feedback to help improve the quality of the paper. We would also like to thank Rajit Kambo, Priya Sethuraman, Shivani Gupta, Adwait Sathye and Nitin Saxena for their support of this work. Finally, we would like to thank Julie Brodeur for editing the paper.

