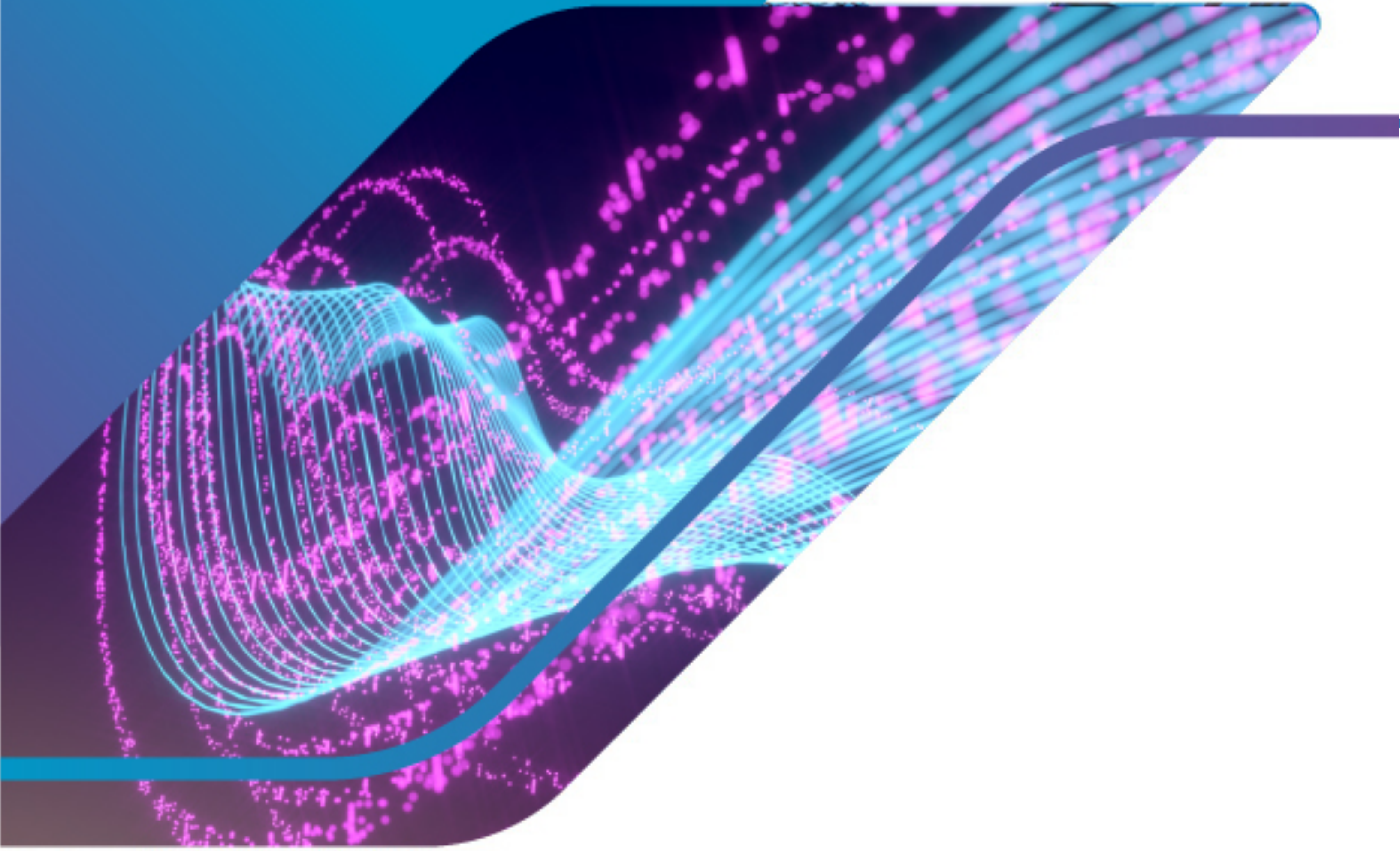
PERFORMANCE TECHNICAL WHITE PAPER
February 2012



VMware vCenter 5.1 Database Performance Improvements and Best Practices for Large-Scale Environments

vmware by Broadcom

Table of contents

Executive summary

VMware vCenter® 5.1 introduces significant improvements to the statistics subsystem. Statistics data represents the largest storage impact in a vCenter database; therefore, vCenter must handle this data in an effective way so that it does not hamper VMware vSphere® performance. To help meet this need, vCenter 5.1 reduces the resource overhead of the database through enhancements to 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 take advantage of the improvements. It shows how to:

- Lay out disks for both Oracle and SQL Server
- Update index statistics for highly volatile tables
- Separate tables and indexes for improved performance
- Take advantage of features in enterprise editions of SQL Server
- Tune certain parameters for Oracle and SQL Server; for example, the parallelism threshold for SQL Server

The improvements are especially important for vCenter 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 can manage.

Introduction

vCenter provides a centralized way to control and monitor virtual infrastructure. It persists important information to a relational database. This data falls into four categories:

1. Inventory and configuration data
2. Task and event data
3. Alarm data
4. Statistics data

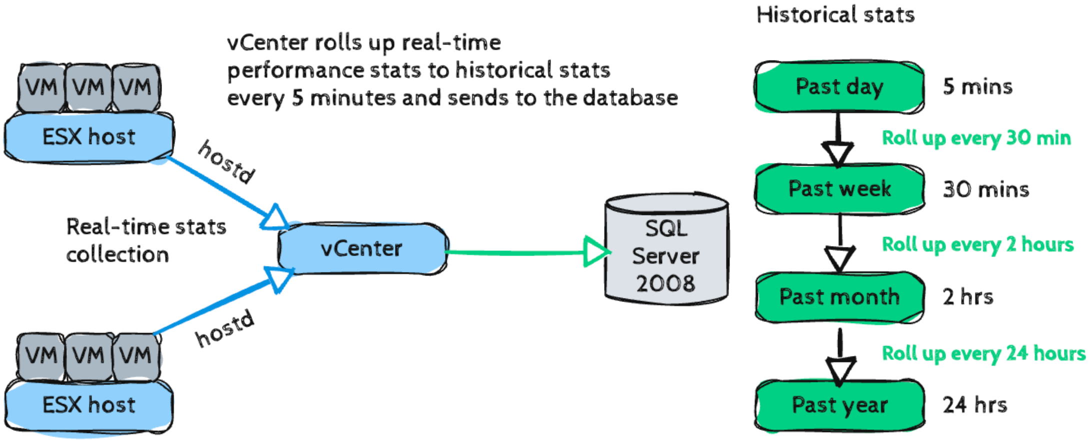
The database is therefore a critical component of vCenter performance. Because the statistics data consumes a large fraction of the database, proper functioning of statistics is an important consideration for overall database performance. Thus, statistics collection and processing are key components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

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

Overview of the statistics subsystem in vCenter

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

Figure 1. Diagram of the statistics subsystem in vCenter



Each VMware ESX host is connected to a vCenter, and the vCenter is connected to a relational database. vCenter collects statistics from each ESX host periodically and persists this data to the relational database. The database in turn runs stored procedures that summarize this data at intervals.

Each ESX host collects statistics at a 20-second granularity. In vCenter, these are called **real-time statistics**. You can view real-time statistics through the vSphere Client by going to the **Performance** tab and clicking the **Advanced** button. The client always receives real-time statistics directly from the ESX host. This ensures the timeliness of the data and doesn't stress the database.

Periodically, these 20-second statistics are aggregated into 5-minute statistics. vCenter stores these 5-minute historical statistics in the past day table. We call the procedure of converting a set of fifteen 20-second real-time statistics into a single 5-minute historical statistic a **rollup**.

The vCenter database stores a variety of statistical granularities. 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:

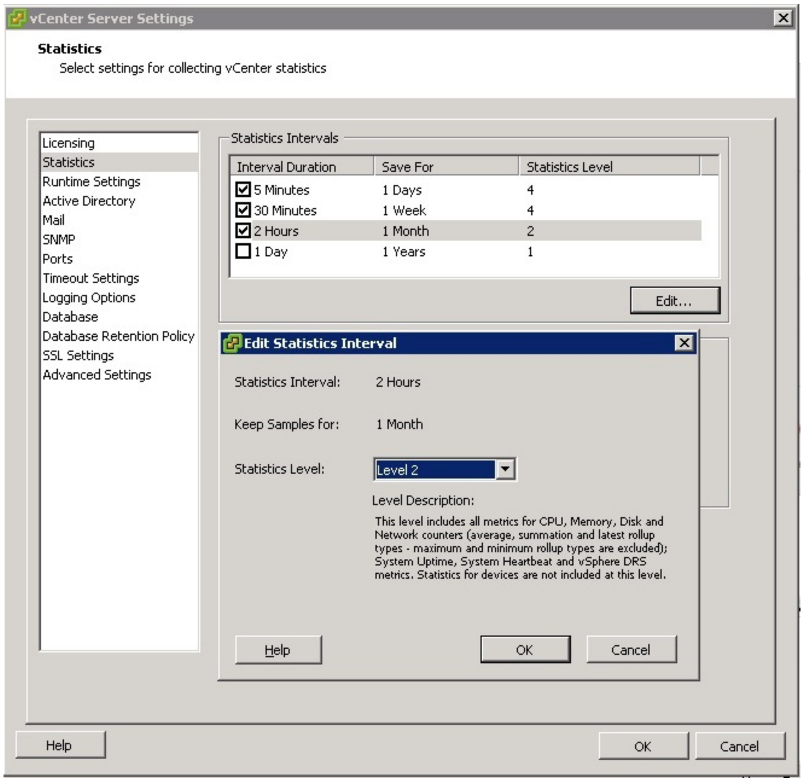
- The **past day** statistics rollup procedure runs every 30 minutes to aggregate the 5-minute historical statistics into 30-minute historical statistics.
- The **past week** statistics rollup procedure runs every 2 hours to aggregate the 30-minute historical statistics into 2-hour historical statistics.
- The **past month** statistics rollup procedure runs every day to aggregate the 2-hour historical statistics into daily historical statistics.

We refer to all statistics stored in the database as **historical statistics**. You can view the historical statistics through the vSphere Client by going to the **Performance** tab, clicking the **Advanced** button, 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: 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:** To reduce the volume of the statistics data going into the database and to allow control of this volume, vCenter has statistics collection levels, which vary from 1 to 4. In general, the higher the level, the more detailed the statistics and therefore the higher volume of 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



Interval Duration	Save For	Statistics Level
☑ 5 Minutes	1 Days	4
☑ 30 Minutes	1 Week	4
☑ 2 Hours	1 Month	2
□ 1 Day	1 Years	1

Statistics Interval: 2 Hours

Keep Samples for: 1 Month

Statistics Level: Level 2

Level Description:

This level includes all metrics for CPU, Memory, Disk and Network counters (average, summation and latest rollup types - maximum and minimum rollup types are excluded); System Uptime, System Heartbeat and vSphere DRS metrics. Statistics for devices are not included at this level.

Table 1. Description of statistics collection levels

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, including maximum and minimum rollup types.

Database performance improvements

This section describes improvements to the statistics subsystem in vCenter 5.1. vCenter collects and maintains a huge 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:

- Reduction in the 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 made possible through three optimizations:

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

Removed staging tables

vSphere 4.1 and vSphere 5.0 used staging tables, which accommodated the bursty behavior of statistics collection on large-scale infrastructures. There were three staging tables:

1. One staging table was used for vCenter to insert the 5-minute statistics. After a fixed time interval, it switched over to the next staging table.
2. The full staging table was parsed concurrently, and all of the 5-minute statistics were inserted into the past day table.
3. The third staging table was used as an extra buffer to make the transition smooth between staging tables.

The demands of the larger inventories supported by vSphere 5.1 required a more scalable solution. To address this issue, vSphere 5.1 removes these staging tables and instead partitions the statistics tables. With this change, vCenter now inserts 5-minute statistics directly into the past day table. This change significantly improves the overall statistics collection system.

The removal of staging tables also provides for more robust statistics data retention for large-scale environments with certain network and storage configurations that experienced some loss of statistics data. By getting rid of staging tables, the logic for inserting statistics can be redesigned. This lowers the database's resource requirements and increases the number of statistics that are saved to the database at once, making vCenter more scalable.

Partitioned statistics tables

There were three sources of I/O to the statistics tables in vCenter:

1. Inserting statistics

2. Rolling up statistics between different intervals

3. Deleting statistics when they expired.

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 table could grow very large with at-scale inventories.

vSphere 5.1 includes redesigned and partitioned statistics tables, which reduce contention and improve performance. With partitioned tables, each table is segmented into separate areas that can be accessed at the same time, reducing the time it takes for multiple vCenter operations to read and write to each table. The sub-tables hold the information for a brief period before condensing and moving it to the next level table.

Table 2. 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 of statistics.

Table	Time interval for each sub-table	# of Time IDs in each sub-table	Time duration for statistics
Past Day	30 minutes	6	5-min stats
Past Week	2 hours	4	30-min stats
Past Month	1 day	12	2-hour stats
Past Year	10 days	10	Daily stats

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 client datacenter and cluster charts, it is possible to see the top 10 virtual machines organized by CPU utilization. This chart is computed by TopN queries that do some math on the statistics to determine the “top N” virtual machines by CPU utilization, 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.

Results of testing with an at-scale inventory

As a result of the changes mentioned earlier, vSphere 5.1 offers these enhancements:

- Insertions have greatly improved.
- Rollup performance has improved significantly. The rollup process is now well-optimized, ensuring that its performance improves with the level of statistics. Additionally, it is very reliable, as shown in Table 3, where the rollup takes only a few minutes.
- Purge performance has greatly improved, nearly eliminating all disk I/O. Removing the expired statistics is now a simple truncation of a sub-table when its data is no longer valid. This reduced the purge time to less than a second.
- Higher statistics levels are now more efficient than they were before.
- We designed tests to demonstrate how many statistics could be collected at various time intervals on a system managed by vCenter with a large inventory.

At-scale inventory specifications

The lab environment used the following at-scale inventory, where the vCenter inventory was very large and met 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 configuration

We used the following virtual machine setup to support the at-scale inventory:

- Virtual machine running vCenter:
  - 16 vCPUs
  - 52GB memory
- Virtual machine running the database (Microsoft SQL Server 2008):
  - 16 vCPUs
  - 26GB memory
- Database files: RAID-5 storage backed up by 10 disks, with each disk having 2Gbps, 5200rpm, and 133.68GB
- Log files and tempdb: 10 disks RAID-5 storage, with each disk having 2Gbps, 5200rpm, and 133.68GB

Number of statistics collected per hour and per five minutes with I/O activity

Table 3 shows the number of statistics that can be inserted per hour for different statistics levels. For example, 80 million statistics per hour were collected at level 4 for the at-scale inventory. 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. See Table 1 for a description of these statistics.

Table 3. The number of statistics that vCenter collects and pushes into the database for different levels for the at-scale inventory

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

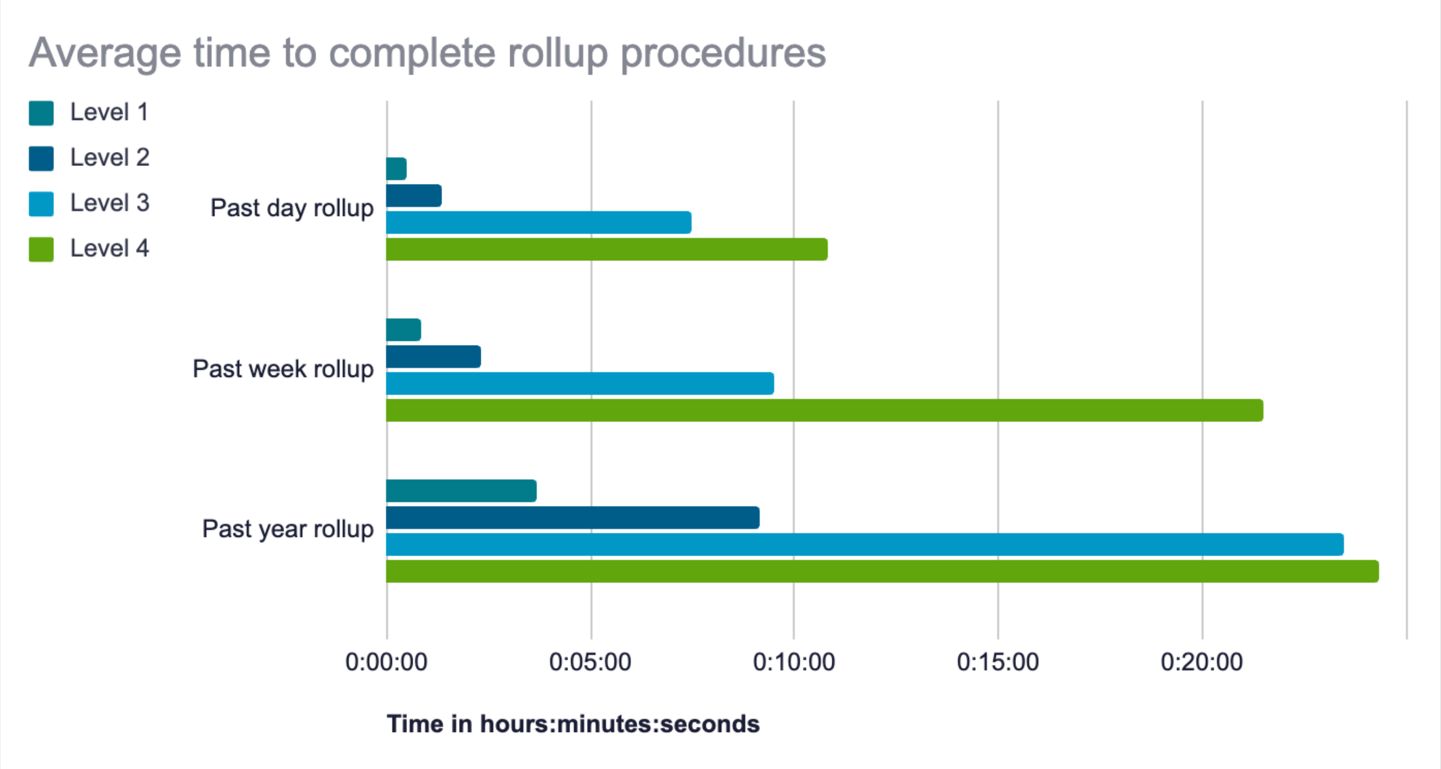
Rollup performance

Table 4 and Figure 3 show how much time it takes to roll up the statistics collected at different levels for the at-scale inventory.

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

	Time (minutes:seconds)			
Type of rollup	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

Figure 3. Chart comparing average time to complete rollup procedures for different statistics collection levels on SQL Server



Database best practices for large-scale environments

This section provides tips for configuring the database to best take advantage of the improvements in vSphere 5.1, including:

- How to lay out disks for Oracle and SQL Server
- How and when to recompute statistics on various tables
- How to separate tables and indexes for improved performance
- How to tune certain parameters for Oracle and SQL Server; for example, the parallelism threshold for SQL Server.

For general vCenter best practices, refer to the Performance Best Practices for VMware vSphere 5.1 guide.

How to lay out disks for SQL Server and Oracle

vCenter operations may generate significant I/O on the database server. We suggest distributing the LUNs/disks evenly and organizing the database files in this manner.

SQL Server

We recommend 4 disks:

1. `mssql01` for the primary and msdb databases(`.mdf` and `.ldf`)
2. `mssql02` for the tempdb files (`.mdf` and `.ldf`); also set the initial size to 10GB
3. `mssql03` for the VCDB files (`.mdf` and `.ldf`)
4. `mssql04` for the VCDB backup location

Oracle Database

We recommend 7 disks:

1. `/u01` for `system01.dbf` and `undotbs01.dbf`
2. `/u02` for `sysaux01` and `temp01.dbf`
3. `/u03` for `vpxdata01.dbf`
4. `/u04` for `vpxindx01.dbf`
5. `/oralog` for `redo01a.log`, `redo02a.log`, and `redo03a.log`
6. `/oralog_mirror` for `redo01b.log`, `redo02b.log`, and `redo03b.log`
7. `/oraarch` for the archive destination

How to update index statistics for highly volatile tables

SQL Server

The cost-based optimizer (CBO) in SQL Server uses statistics about tables and indexes to compute the most efficient access plan. In SQL Server, the database option `AUTO_UPDATE_STATISTICS` automatically updates index 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 collected 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 must be inserted, updated, or deleted before the statistics are automatically updated. This could cause problems for vCenter operations.

A few tables in the vCenter schema change data at a very rapid rate based on certain vCenter 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 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 as follows:

- To update statistics for the database, use:

```
sp_updatestats VCDB;
```

- To update statistics on a table, use:

```
UPDATE STATISTICS TABLE_NAME;
```

Replace `TABLE_NAME` with the actual table name, for example:

```
UPDATE STATISTICS VPX_PROPERTY_BULLETIN;
```

Oracle Database

The cost-based optimizer (CBO) finds the best plan for accessing data, but it relies on having up-to-date statistics. Outdated statistics may negatively impact the database response. The Oracle (10g and 11g) database is set by default to automatically gather statistics. Automatic optimizer statistics collection is usually enough for most database objects that are modified at a moderate speed. However, there are some situations where this collection is not sufficient. This is because the statistics collection job runs during maintenance windows, and data on very large tables can change rapidly. The data in these tables becomes outdated quickly.

The vCenter database content changes rapidly during certain vCenter operations. We recommend gathering statistics regularly to ensure they accurately represent the characteristics of database objects.

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

When the statistics for a table or index are updated, Oracle invalidates any SQL statements that are currently parsed and accessing that table or index. However, the next time a similar SQL statement runs, it is re-parsed, and the optimizer automatically selects a new execution plan based on the updated statistics.

- 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 and highly transactional databases, try moving non-cluster indexes and tempdb onto their own file groups. We did not test this option, but we believe it will work.

> **Warning:** Because 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 Database

In addition to following the recommended disk layout previously described, we recommend separating indexes from the `/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 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. This option determines the computing and thread resources used for parallel query processing.

➢ To set the max degree of parallelism:

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

Where `n` is the number of processes.

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 this value.

➢ 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 database that involved redesigning the tables and revising the stored procedures for the statistics subsystem. With the help of these changes, vCenter 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
- Can 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.

About the authors

Alper Mizrak is a software engineer in Broadcom's VMware Cloud Foundation (VCF) Performance team. His focus is on the performance and scalability of VMware products. Alper earned his PhD in computer science from UC San Diego in 2007, specializing in network security and distributed systems. His graduate research earned the prestigious William C. Carter Award and resulted in a book, *Secure Networking: Detecting Malicious Routers* (2008), along with several academic publications and technical reports.

Raju Angani is a software engineer in Broadcom's VCF Performance team. In this role, he has focused on vCenter database design, optimization, and performance tuning. He has written and contributed to several other technical papers, including the “VMware vCenter 8.0 U3 Tagging Best Practices” paper.

Acknowledgments

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.