Management and maintenance of the Orion platform database is covered in this paper and this paper is only applicable to products using this database.
Introduction

This paper will help you understand how to best manage the version of SQL you are using in your production IT environment. Many SolarWinds Products use a Microsoft® (MS) SQL database for storing and retrieving data. These products include:

- Network Performance Monitor (NPM)
- Server and Application Monitor (SAM, formerly APM)
- NetFlow Traffic Analyzer
- VoIP and Network Quality Manager (VNQM)
- IP Address Manager (IPAM)
- Synthetic End User Monitor (SEUM)

When you install any of these products, you have the choice of linking them either to an existing copy or new copy of SQL Server. For convenience, all evaluation downloads come with a copy of MS SQL Express, Microsoft’s free version of MS SQL.

This paper only covers the use of MS SQL as it pertains to a SolarWinds Orion installation and is not intended to provide a comprehensive guide to MS SQL management. Step-by-step instructions for accomplishing tasks in MS SQL are not covered here because of the differences in MS SQL tools across SQL versions. Guidelines for what the tasks are intended to solve are included with external references for instruction.

SQL Installation and Architecture Options

The standard SQL environment for Orion products contains:

- A dedicated SQL Standard or Enterprise Server
- Directly attached (DAS), RAID 10 storage (I/O subsystem)
- LAN attachment between the main Orion server and any additional components.

![Diagram of SQL installation and architecture options](image)

SolarWinds recommends the above deployment as it has proven consistently to be the best performing solution.

You may need to depart from this design to comply with your IT standards. Below are some of the options used and the risk factors associated with each. Altering any part of the SQL environment is potentially troublesome. Compromised SQL performance, and thus compromised Orion product performance is the most common effect of a poorly executed SQL installation.
Option 1 – Virtual Orion Server(s)

You may choose to install Orion products on virtual servers. The requirement for virtual servers is that they be assigned the same resources as a physical server would have. Specific requirements for servers are listed in the product’s Administrative Guide. This includes additional poling engines, additional web servers, and Failover engines. Using virtual servers for Orion is only if the virtual servers are undersized.

Option 2 – Virtual SQL Server

Depending on the expected load for your SQL server, a virtualized SQL server may be a viable option. Performance can degrade if the host server becomes over provisioned with virtual servers and the virtual SQL server is unable to access the host resources.

Option 3 – Shared SQL Server

For small to medium sized deployments, up to the 2000 element license level, you may decide to install the Orion database as an instance on an existing SQL server. However, for large Orion installations, those at the 2000 or unlimited element size, you should always use a dedicated SQL server. If an existing server is to be used, verify that it meets all the CPU, memory, and storage requirements of a dedicated server. Also be aware that sharing a SQL server may introduce performance issues caused by other systems that are also using the SQL server.

Option 4 – Shared SQL Storage

Using an existing shares storage platform is attractive from a cost point of view. Several SolarWinds Orion implementations use this option. This typically consists of a Storage Area Network (SAN) and a high density, shared storage platform. Poor storage performance can have a significant impact on Orion performance. At a minimum you should make sure that the LUN is on a RAID 10 array. One common mistake is to place the storage on a Storage Area Network (SAN) that is RAID 5, RAID 6 configured or has performance lower than a DAS subsystem. RAID 5 and Raid 6 configurations are not supported and should never be used for Orion MS SQL storage.

Acceptable numbers for storage are disk write latency of 1-5 ms or less for data (mdf/ndf) files and 20 ms or better for log (ldf) files. The throughput of fibre channel is approximately 200 MBps per Gpbs. Using an I/O testing tool, such as IOMeter®; you can measure the throughput and latency of any storage. If you are able to demonstrate SAN performance is equivalent to DAS performance and is not bandwidth restricted, this solution may be an option.

Maximizing SQL Server Performance

Making the proper decision during installation is critical to the overall performance, stability, and expandability of the Orion products. You should consider the following factors:

- Which Orion products will be using this database?
- How many elements do you expect to be monitoring?
- How long must you retain historical data?
- If NetFlow Traffic Analyzer (NTA) is installed, does it store flow data in the SQL database or in the dedicated NTA Flow Storage Database? If flow data are still stored in the SQL database (NTA 3.11 and older, NTA 4.0 installed on 32-bit operating systems), how many interfaces will be configured to export flow data?
- How many Server and Application Monitor (SAM, formerly APM) monitors do you expect to need?

Understanding these factors and the load they will place on the SQL server will help you properly size the server. Some of the following rules of thumb will also be helpful:
• SQL Express is only suitable for very small Orion installations that do not have NTA.
• WAN connections should never be used between the SQL server and the Orion server(s). This includes any additional Orion pollers used.
• SQL Server should not be installed on the Orion server.
• NetFlow can be a major factor on database sizing, depending on the incoming flow rates.
• The performance of a SQL server is dependent on the performance of the I/O subsystem.
• The more disks in a RAID 10 array the better.
• Many RAID controllers do not handle RAID 01 well.
• It can be very difficult to upgrade an underpowered SQL server in a production deployment.

Taking these factors into consideration, we can design the SQL server for maximum performance. Below are the best practices for creating a well performing SQL server.

File Segregation

Separating the data files from the log files in storage can boost performance significantly. The SolarWinds Orion database consists of data files with .mdf or .ndf file extensions, and log files with an .ldf extension. These files exist for the main database and for the temporary (temp) database SQL uses for moving data.

In small to medium installations without NetFlow, this can be accomplished by placing the files on separate physical drives. This configuration would look like the below:

C: = OS and SQL Server, 2 drives RAID 1
E: = SQL data files, 2 drives RAID 1
F: = SQL log files, 2 drives RAID 1

An additional option would be to place the temporary database file on two additional drives.

The need for a high performance storage solution performance increases when you are using NTA 3.11 or older, or NTA 4.0 on 32-bit operating systems where flow data are stored in the SQL database. A SQL storage system for NPM and NTA (with or without additional products) will perform well when the same separation of files as seen above is augmented by RAID 10 arrays. This is shown below:

C: = OS and SQL Server, 2 drives RAID 1
E: = main data files, 6 drives RAID 10
F: = main log files, 3 drives RAID 0
G: = temporary data files, 6 drives RAID 10
H: = temporary log files, 3 drives RAID 0

Drive Speed

Drive performance and drive speed are not linearly proportional. You will realize about at 20 – 25% increase in RAID throughput by installing 15,000 RPM hard drives instead of 10,000 RPM drives.

Solid State Drives (SSDs) offer read and write speed approaching that of volatile memory. SSDs offer read and write throughput speeds about twenty times that of 15,000 RPM spindle hard drives. SSDs can have a shorter life expectance than spindle drives, especially in a read/write intensive application such as SQL.
Memory

Install the 64-bit version of SQL. The 64-bit version can access memory above the 4GB 32-bit limit. With additional memory and the proper software, it is possible to mount the temporary volumes in RAM as emulated drives. Temporary databases will be approximately one forth the size of the main database files.

You should limit the memory SQL can access to leave reasonable space for the OS and other programs. SQL Server can be configured to reserve memory for SQL processes. If not limited, SQL will reserve all of the available system memory. This creates performance issues when the OS needs more memory but SQL has reserved it. The OS has no choice but to page memory and this will bring the server to a standstill. Consult your OS and SQL version documentation for guidelines in this adjustment.

CPU

64-bit SQL takes full advantage of multiple core CPUs. For best performance, disable all CPU power saving tools.

Sizing Storage Capacity

Your Orion database will be somewhere between 20 GB and 450 GB. Some guidelines that will help you properly size the SQL storage are listed below:

- An installation only with SolarWinds NPM will not use more than the recommended minimum of 20 GB of SQL storage.
- If you are running SolarWinds NTA 3.11 or older, or NTA 4.0 on a 32-bit operating system, NTA stores flow data in the SQL database. NTA will thus be the largest single factor in database size. In very large SolarWinds NTA installations (3000+ exporters) the SQL database can be as large as 450 GB.
- Lengthening any of the data retention settings may have a large effect on database size.
For general guidance in database sizing, SolarWinds pre-sales engineering can assist evaluation installations. Support can assist with licensed installations that are under an active maintenance agreement. These sources can approximate sizing, but sizing is a multidimensional and complicated task. Every environment is somewhat unique so measuring the actual load of the SQL database will be very helpful in understanding the size requirements.

**NTA 3.11 and Older, NTA 4.0 on 32-Bit Operating Systems: Estimating the Orion Database Growth due to NTA Sources**

If you are using SolarWinds NTA 3.11 or older, or NTA 4.0 on a 32-bit operating system, flow data are stored in the SQL database. NTA will probably be the main factor in database growth, and you can use modeling the database size for a better estimation of the database size requirements.

**Note:** NTA 4.0 on 64-bit operating systems started storing flow data in a dedicated database, the NTA Flow Storage Database. Starting from NTA 4.1, NTA Flow Storage Database will be the only supported storage for flow data, and NTA will no longer be the single largest factor affecting the Orion database requirements. However, NTA keeps using the SQL database for storing CBQoS data.

To estimate the growth of your Orion database due to adding NTA sources, follow these steps:

1. Ensure that you have a stable Orion NPM/NTA server installed and collecting data for at least 5 days.
2. Launch the SolarWinds Database manager in All Programs -> SolarWinds Orion -> Advanced Features select Database Manager, and then click Add SQL SERVER.
3. Type in the IP address and SQL authentication for your database.
4. Right click on your database and select New Query. Copy and paste the following query:

   ```sql
   SELECT RTRIM(name) AS [Segment Name], groupid AS [Group Id], filename AS [File Name],
   CAST(size/128.0 AS DECIMAL(10,2)) AS [Size in MB],
   CAST(FILEPROPERTY(name, 'SpaceUsed')/128.0 AS DECIMAL(10,2)) AS [Space Used in MB],
   CAST(size/128.0-(FILEPROPERTY(name, 'SpaceUsed')/128.0) AS DECIMAL(10,2)) AS [Available Space in MB],
   CAST((CAST(FILEPROPERTY(name, 'SpaceUsed')/128.0 AS DECIMAL(10,2))/CAST(size/128.0 AS DECIMAL(10,2)))*100 AS DECIMAL(10,2)) AS [Percent Used]
   FROM sysfiles
   ORDER BY groupid DESC
   ```

   You should see a table similar to the one below.
This view gives a lot of valuable information about the database from a high level. Here you can see:

- The total size of the database
- The available free disk space
- Details about the database files
- These files are the data file (.mdf), the log file (.ldf).
- From the displayed statistics, we can see that the Orion DB size is 131.25 MB, there are 43.19 MB of used space and 88.06 MB free.

**Note:** In NTA 4.0 64-bit and in newer NTA versions, all file groups (FG1 - FG4) will be almost empty.

If we do not need to add more devices, SolarWinds SAM monitors or SolarWinds NTA flows, this database can be expected to perform well and need no additional space.

But what if you want to add additional SolarWinds NTA flow exports to this database? By examining the pertinent NTA of the tables in the SolarWinds Orion database, you can estimate the additional storage requirements.

The most critical tables for NTA database growth are:

- NetFlowDetail_Y_XXXXXXX, where Y is the node ID of the exporter and XXXXXXX is the decimal time period start.
- NetFlow EndPoints where the IP address of every observed endpoint is stored.
- NetFlowSummary1 which contains the results of the first data roll up (1 min – 15 min).
- NetFlowSummary2 which contains the second level roll up (15 min – 1 hour).
- NetFlowSummary3 which contains the final roll up (1 hour – 24 hours)

These tables are shown below using the Orion Database Manager.
The below graph shows the database roll up intervals.

For data collected in the last hour, SolarWinds NTA saves the data in 1 minute intervals. At point A, NTA compresses the flow data into 15 minute periods. This provides a 15:1 data reduction. After 24 hours the 15 minute data is compressed to hourly data points (point B). This provides a 4:1 data reduction. After 3 days the hourly data is compressed to daily data points (point C). This provides a 24:1 data reduction. Point D is the maximum data retention setting where data is discarded. By default this is 30 days. Point A and D are configurable by the Orion Administrator. Points B and C are not configurable. The Database Settings grouping in NTA 3.11 or older, or in NTA 4.0 on 32-bit operating systems is shown below:

Below are the factors you can use to estimate growth of these tables.

- NetFlowEndpoints. This table is large but typically unchanging in size. When adding several new flow sources, allow for a 20% growth in this table.
- NetFlowDetail_Y_XXXXXX. The last 15 minutes of raw data is stored in these tables. NTA creates five of these tables for each node. Y represents the node ID and XXXXXX represents the time interval. When NTA summarizes the data from the oldest of these tables, it deletes that table and creates a new one. These represent the aggregated flow collection for all interfaces on a device. Adding a new NTA interface to an existing NTA source device expands this table by the relative size and rate of the new interface. Adding a new device as a NetFlow source will add a new raw NTA table. To estimate the effect on the database growth, locate a node with similar NetFlow interfaces. Use the size of the detail tables associated with the similar node to approximate the new raw NTA data table. Multiple these results by five to allow for the five retained intervals.

- The NetFlow Summary 1 table grows by the relative size of the new NetFlow sources you add.

- The actual current database size is seen by right clicking on the table name in Orion Database Manager and choosing Database Details. Be sure to sum the index and data sizes as listed.

Assuming the NTA system has been running for at least 30 days, the same calculations can be used to approximate the growth of Summary2 and Summary 3 tables.

**Notes:**

The absolute storage numbers in this example are completely hypothetical and should not be interpreted as the storage required for you SQL server storage.

The temporary database will be about 25% of the space of the main database. Add 25% onto any calculated growth number to allow for temporary database growth.

**NTA 4.0 on 64-Bit Operating Systems and Newer Versions**

NTA 4.0 on 64-bit operating systems does not use the Orion SQL database for storing flow data. Flow data are stored in the NTA Flow Storage Database instead.

**Estimating the NTA Flow Storage space necessary for NTA 4.0 on 64-bit operating systems and newer**

For more information about estimating the NTA Flow Storage Database size on NTA 4.0 on 64-bit operating systems and newer, see “How do I determine the right amount of CPU/Mem for my NTA Flow Storage Database” in the SolarWinds knowledge base.

**Estimating the SQL database space required by NTA 4.0 on 64-bit operating systems and newer**

In NTA 4.0 on 64-bit operating systems, NTA uses the SQL database only for storing CBQoS data. As a result, NTA does not need so much space as in previous NTA versions. CBQoS data grow in a linear way compared to the total amount of monitored CBQoS nodes.

**To estimate the space needed for individual CBQoS nodes:**

1. Find out see the size of the following CBQoS tables in the SQL database:
   - cbQoSClassMap
   - cbQoSPolicy
   - cbQoSPolicyMap
   - cbQoSStats.

2. Sum up the space taken up by these tables and divide it by the number of CBQoS nodes. You will get the average space required for each additional CBQoS node.
Best Practices for Managing the Orion Database

As your SQL database matures or after adding new Orion products, the database may become larger than you originally estimated or might slow unexpectedly. Several factors may cause these issues. This section explores the most common issues and explains how to correct the issues.

Managing Database Growth in the Orion Web Interface.

The most common issues with SolarWinds Orion databases are related to the database size. Properly managing size can help you avoid issues with storage capacity and database performance. A primary factor in database size is the data retention settings available in SolarWinds Orion. Each SolarWinds Orion Product allows you to manage the data rollup periods and the data retention limit. The impact of adjusting any of the data retention and rollup windows will be roughly proportional to the effect the Orion Product has on the database size. When considering expanding a data retention period, you may be able to make small changes and examine the impact on size and performance as you approach the desired new limit.

Not all of the data rollup periods are adjustable. Typically you can alter one intermediate data rollup period and the limit for data retention. The database options for SolarWinds NPM are shown below:
Each of the above settings are at the default level. The relative impact of extending any of the data retention can be summarized by this rule: **The shorter the data interval, the greater the effect the setting will have on the database size.**

- Extending the detailed data retention will have the largest potential impact on database size and performance.
- Extending hourly retention will have a lesser effect.
- Extending daily retention will have the least effect.

This is due to the summarization of detailed data into hourly data increments and then into daily data increments. Each SolarWinds Orion Product allows similar data retention options and the above guidelines should be followed for each product.

**Database Maintenance**

A database maintenance routine is built in to each SolarWinds Orion product. Database maintenance is enabled by default. Maintenance routines vary the products installed. Because of the nature of SQL, this works well in small to medium deployments.

We realize not every SolarWinds Administrator is an expert in tuning and managing the SQL database. As the database architecture changes with various versions of SolarWinds Orion products, we occasionally alter the database maintenance routine to assist you in managing the database. Database maintenance routines are often product-specific.

As previously stated, NTA 3.11 or older versions and NTA 4.0 on 32-bit operating systems store flow data in the Orion database. NTA is thus the largest driver in database size, performance and complexity. While SolarWinds does not directly support custom maintenance plans, if you have a large database (200GB+) with NTA, you should consider creating your own maintenance plan. SQL maintenance plans should be created with the assistance of a knowledgeable SQL administrator. In general, your plan should contain the following elements:

- An analysis of white space within the database files. This is analogous to data fragmentation.
- A general data integrity check.
- Detection of index fragmentation. This causes index searches to slow or fail.
- A re-indexing routine.
- No auto grow or auto shrink.
- A tested backup and restore plan.

**Note:** The default database maintenance routines in SolarWinds Orion products do not contain a backup or restore plan.

SQL auto grow and auto shrink can contribute to unnecessary database tasks and index fragmentation. Using auto grow and auto shrink is like having your house expanded to accommodate guests for a holiday, then having the expansion removed when the guests depart, then expanding and contracting for the next occasion. The better solution is to have a location that has been correctly sized for guests and leave it that way. When a SQL database grows and shrinks, the index becomes fragmented and may point to data that no longer exists. SQL indexes can become so fragmented that it hinders searches.
To deal with index fragmentation problems:

- Do not use auto grow or auto shrink whenever possible.
- Do not manually shrink your database to recover disk space.
- Include re-indexing in your maintenance routine.

A common error in managing a database is to shrink the database that is running out of disk space. This can have devastating results. Keep in mind that the index is a file on your database drive. If you shrink a database with insufficient space to update index files, the index may become so fragmented and incomplete that indexed searching is not possible. When this happens, your Orion installation will constantly timeout in the web console.

A better solution for addressing database storage that is critically low on free space is to back up the database, add additional storage, and then restore the database. Another possible solution is to prepare a new, larger storage solution, detach the Orion database from the old storage, copy the database to the new storage, and then reattach the database on the new storage.

The best possible solution to avoiding these emergencies is to build the database large enough from the beginning and carefully monitor its growth and storage over time.

**Troubleshooting an Orion Database Issue**

Two of the most common symptoms of database issues are degraded Orion performance, and errors related to the inability to connect to the database.

**Notes:** This section covers the basics of determining a database issue as the issue pertains to interaction with Orion.

This is not intended to be a comprehensive MS SQL troubleshooting guide.

In the Orion database, the single most important SQL server performance measurement is disk queue length. Queue length is a measurement of the SQL writes that are waiting to be written to disk. When disk queues start lengthening and there is a steady load on the SQL writes, the queues may grow so large that write requests get dropped. This may lead to gaps in data and will affect the overall performance of the SQL server. A good rule of thumb is that disk queue length should not exceed two times the number of effective spindles in the SQL storage. The effective spindle count is the number of striped spindles. For a RAID 10 direct attached storage unit with eight total disks, the effective spindle count is four. Four of the spindles in this array are the primary striped array and the other four are a secondary striped mirror of the four primary spindles. Since no performance gain is achieved by mirroring disks, only the primary striped set is used to measure performance.

For additional information on database performance see the [Managing Orion Performance Technical Reference](#).

When errors occur that point to a loss of the connection to the database the following steps can help isolate the issue:

1. Ping the SQL server from the Orion server to check network connectivity.
2. Open SQL Server Management Studio or the Orion Database Manager and attempt to connect to the database.
3. If both of the above are successful, run the Orion Configuration Wizard against the database by selecting Database in the first wizard screen. Ensure that you are using the proper database credentials.
4. Open the Orion web UI to test connectivity again.
5. Test opening an ODBC connection from the Orion server using a Microsoft utility such as ODBCPing
   http://support.microsoft.com/kb/138541.

   If all of this fails, then the issue is a failure with the SQL server. At this point you will need to go directly to
   the SQL server and begin troubleshooting. Troubleshooting SQL is very specific for each version and
   implementation. For Microsoft guides to SQL troubleshooting see
   alog=LCID%3D1033&mode=r.

**Additional Resources**

SQL Index Fragmentation
alog=LCID%3D1033&mode=r

http://knowledgebase.solarwinds.com/kb/questions/3202/Critical+index+fragmentation+detected+during+
database+maintenance

http://knowledgebase.solarwinds.com/kb/questions/790/Netflow+database+maintenance

http://knowledgebase.solarwinds.com/kb/categories/Orion+NetFlow+Traffic+Analyzer+%28NTA%29/Data
base+%28NTA%29/

SQL Index Rebuilding


SQL table and Index Structures landing page


SolarWinds thwack Community

www.thwack.solarwinds.com