

SOLARWINDS TECHNICAL REFERENCE

Managing Orion Performance

Orion Components Overview	1
Managing Orion Component Performance	3
SQL Performance - Measuring and Monitoring a Production Server	3
Determining SQL Server Performance Abilities	5
Optimizing SQL Server for Orion	6
Orion Server Performance	8
Appendix A – Other Resources	10

This paper examines how to measure and optimize the performance of an Orion installation.

Copyright © 1995-2012 SolarWinds Worldwide, LLC. All rights reserved worldwide.

No part of this document may be reproduced by any means nor modified, decompiled, disassembled, published or distributed, in whole or in part, or translated to any electronic medium or other means without the written consent of SolarWinds. All right, title, and interest in and to the software and documentation are and shall remain the exclusive property of SolarWinds and its respective licensors.

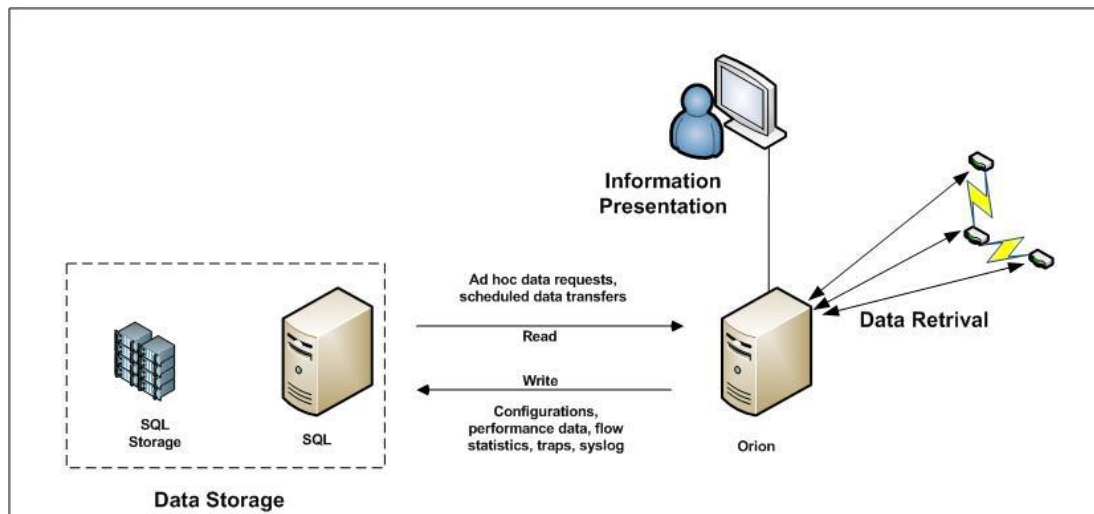
SOLARWINDS DISCLAIMS ALL WARRANTIES, CONDITIONS OR OTHER TERMS, EXPRESS OR IMPLIED, STATUTORY OR OTHERWISE, ON SOFTWARE AND DOCUMENTATION FURNISHED HEREUNDER INCLUDING WITHOUT LIMITATION THE WARRANTIES OF DESIGN, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE, AND NONINFRINGEMENT. IN NO EVENT SHALL SOLARWINDS, ITS SUPPLIERS, NOR ITS LICENSORS BE LIABLE FOR ANY DAMAGES, WHETHER ARISING IN TORT, CONTRACT OR ANY OTHER LEGAL THEORY EVEN IF SOLARWINDS HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

The SolarWinds, the SolarWinds & Design, ipMonitor, LANsurveyor, Orion, and other SolarWinds marks, identified on the SolarWinds website, as updated from SolarWinds from time to time and incorporated herein, are registered with the U.S. Patent and Trademark Office and may be registered or pending registration in other countries. All other SolarWinds trademarks may be common law marks or registered or pending registration in the United States or in other countries. All other trademarks or registered trademarks contained and/or mentioned herein are used for identification purposes only and may be trademarks or registered trademarks of their respective companies. Microsoft®, Windows®, and SQL Server® are registered trademarks of Microsoft Corporation in the United States and/or other countries.

Document Revised: 05/15/2012

Orion Components Overview

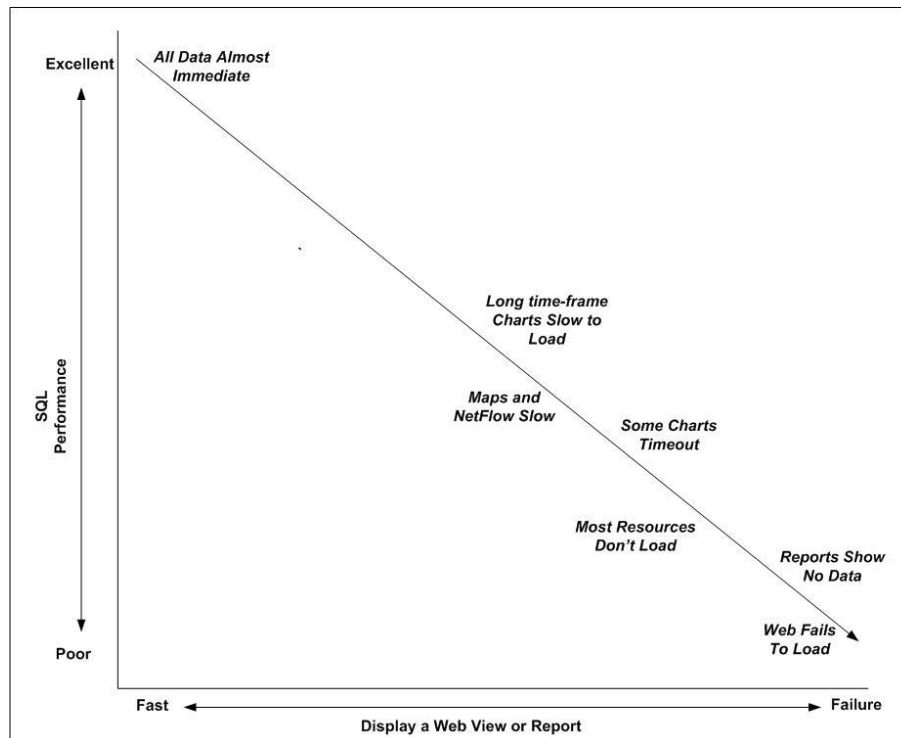
Orion installations can be divided into 3 components, the Orion application server, the MS SQL database and the database storage. These components constantly interact to perform the 3 primary functions of an NMS, data retrieval, data storage, and information presentation. This is depicted below.



Orion Deployment – Major Components and Functions

Each component is dependent on the other components and their functions. For example, in order to view or report on monitored elements, the data must be available from data retrieval, properly stored, and retrieved to the web in a timely manner. If any one of these steps fails, or performs poorly, the desired result will be impacted. The level of impact depends on the performance and availability of all the functions involved. As the performance of any particular function decreases, the level of impact will increase to a point where all the components have failed. Because of the Orion component interdependencies, it can be difficult to determine where the problem lies and how to fix it. This paper will focus on the common causes of poor performance, and provide steps to isolate and rectify the issue.

The SQL data storage is central to most functions of Orion, and so a poorly performing SQL server will greatly affect the user experience. For example, the below graph depicts the impact to Orion functions caused by a poorly performing SQL server.



Web and Report Loading as a Function of SQL Performance

As seen in the above, as SQL performance degrades, a variety of symptoms can result, from slow loading to complete failure. As the symptoms worsen, it can appear that other Orion functions, such as the web engine have failed. Making assumptions about the cause of performance issues or failures can cause wasted time examining components that are performing well. While the root cause of the web failing to load may look like a web failure, in the above, the actual cause is poor SQL performance. For this reason, when a failure occurs it is recommended to follow a step-by-step, documented troubleshooting procedure to isolate the cause of a performance issue.

Performance should be thought of as having a faster-the-better overall goal; performance improvements can be made even though the system runs according to user expectations. Performance is not measured simply as good/bad, but as having varying degrees. Using this proactive approach to performance has many benefits, including:

- Cost avoidance – When the NPM system is optimized, more elements can be added than would be possible in a nominally performing system.
- Outage avoidance – Keeping ahead of performance issues will help to eliminate outages in the Orion system caused by a resource overload.
- More accurate planning – When you know what the system is capable of, you can properly plan for growth without surprises.
- Ease of use – While the Orion web interface is very intuitive, any system experiencing slowness degrades the user experience. Keeping the web interface responsive helps you find and isolate network issues easily and quickly.

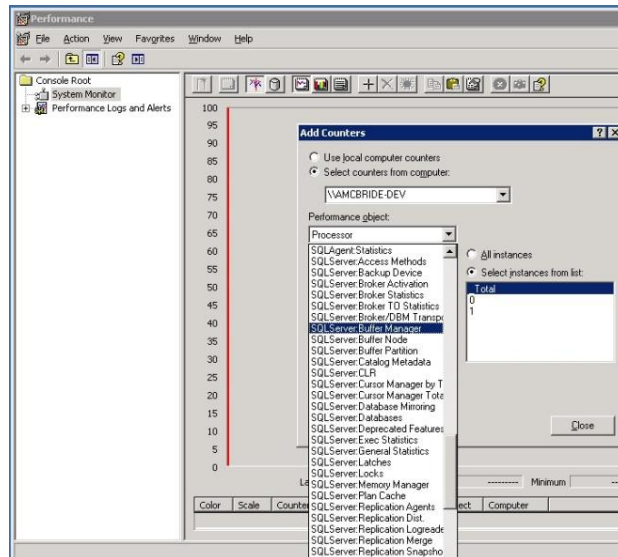
Managing Orion Component Performance

The overall performance of Orion is dependent on the performance of each of the components. Therefore, accurately measuring Orion performance involves measuring the performance of the components as well as the Orion Web Console responsiveness. MS SQL performance is involved in most Orion performance issues, so that is a logical place to begin.

SQL Performance - Measuring and Monitoring a Production Server

Microsoft makes the performance counter accessible through the MS Server Performance Monitor in Administrative Tools menu. This tool makes it easy to locate and observe the counters you will use to measure SQL server performance.

In the Orion environment, the single most important SQL server performance metric 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 snowball to the extent that write requests get dropped. This may lead to gaps in Orion data and will affect the overall performance of the SQL server and Orion. A good rule of thumb is that disk queue length should not exceed 2 times the number of effective spindles in the SQL storage. The effective spindle count is the number of stripped spindles. For a RAID 10 direct attached storage unit with 8 total disks the effective spindle count is 4. 4 of the spindles in this array are the primary striped array and the other 4 are a secondary striped mirror of the 4 primary spindles. Since no performance gain is achieved by mirroring disks, only the primary striped set is used to measure performance.



Keeping in mind that this is a rule of thumb, and that performance is a continuum rather than good/bad assessment, the recommendation does not mean that if you have 5 effective spindles and you peak the queue at fifteen, you will be having performance issues. Also this number should not be used to determine the number of spindles required to lower the queue length. This is because the effect of queue length on performance is not linear. Adding 1 effective spindle to a 4 effective spindle array may lower the queue length by fifty percent or more. The below table provides some other important SQL performance counters and their target ranges.

Metric	Description	Target Range
SQL Server: Access Methods - Page splits/sec	Index or data page is split and partially allocated to a new page. May cause IO issues such as excessive disk queue length. Can be addressed by increasing SQL fill factor.	Smaller the better. A static page splits number with low disk queue length indicates this is not an issue.
SQL Server: Access Methods – Full Scans/sec	Table index failures. When table indexes are not used the result is a disk operation that could have been avoided. High full scan rates will impact transaction rates and overall SQL performance.	Lower the better. Near 0 rates are to be expected for a well performing SQL server.
SQLServer: SQL Statistics - Batch Requests/Sec	A measurement of transaction rate. High batch requests may indicate heavy CPU load.	Static rate. Usual rates are in the thousands and are not a problem unless CPU utilization is too high. Monitor for a slow creep up or any sudden jump that does not recover to normal levels.
SQL Statistics - SQL Re-compilations/sec (re-comps)	A query execution measurement. High re-comp rates will drive CPU utilizations	Lower the better. Should be 0 most of the time.
SQL Server: Locks: Dead locks/sec	Dead lock rate.	0 - dead locks rates higher than 0 should be investigated using MS SQL tools.
SQL Server: Buffer Manager – Buffer Cache hit ratio	The percent of requests that can be served from cache rather than disk. Although writes are more important than reads in an Orion environment, this metric should be tracked as an indicator of memory issues.	Higher the better.
Physical disk – Reads/sec, writes/sec	The read and write rates. This should be measured for logs and data. Because these measure these actual read and write rates they are not an indicator of the maximum I/O rates unless they reach a level where disk queue lengths also increase.	Static levels.

While this is not a complete list of the metrics you may wish to monitor, these do a good job of detecting bottlenecks for disk I/O, memory and CPU. The thwack.com user community has posted several times

Along with monitoring the above SQL counter metrics, some basic metrics for the SQL server should also be monitored. These include the following:

Metric	Description	Recommendation
Database size	Total size of the Orion database. Normally less than 20GB for NPM with moderate syslog data and no NetFlow data. NetFlow can increase the database size sever hundreds of GB. Larger databases require more disk, memory and CPU resources. Large numbers of syslog messages will also increase the database size.	Balance data retention settings with database size and performance.
SQL Server Memory	The amount of RAM used by SQL.	Maximize the memory available to SQL. Increase the minimum SQL memory setting to allocate eighty to ninety percent of the total memory to SQL.
SQL Server CPU	CPU utilization	Keep the average CPU utilization as low as possible. CPU peaks of one hundred percent are acceptable as long as the average stays low and the peaks are brief.
AWE Memory	AWE memory allow for more efficient use of memory on 32-bit SQL.	Enable AWE for all 32-bit SQL. See the Microsoft SMDN Library for details.

Determining SQL Server Performance Abilities

The above section is used for measuring the performance of the SQL server as it operates. You may still want to know the maximum rates that you can expect from your system, this will give you a sense of when your server is reaching its limits. RAM and CPUs operate at predetermined speeds, but storage subsystem performance is dependent on several factors. Several tools can be used to assess the maximum performance of your storage subsystem. Perhaps the 2 most common tools are Microsoft's SQLIO (free) and IO Meter (shareware). These tools give you 2 crucial measurements for your subsystem, the read/write speeds and the disk queue lengths. They will also tell you the maximum throughput rates in MB/s. These tests should be performed on new subsystems in a lab environment, not against an active production server. In general, a well performing subsystem should have disk write speeds of 1 to 5 ms or better for log volumes and 20 ms or better for data volumes. These volumes are discussed in the next section.

Optimizing SQL Server for Orion

Once you understand the important metrics to measure, you can address areas in SQL to maximize performance. Again, as SQL is very storage intensive, this section will focus a great deal on SLQ storage enhancement. Here are some of the basic recommendations for using SQL with Orion. Many of these can be found in the Administrator’s Guide for NPM and the Orion Common Components Administrator’s Guide.

Item	Recommendation
Dedicated SQL Server	Except for small NPM installations (SL250 or lower) with no Orion Modules we recommend a dedicated hardware server be used. While some customers report good performance on virtual servers, a dedicated hardware server will usually result in much better performance.
RAID Array	RAID 10. RAID 10 offers disk write speeds that increase with each spindle (RAID 0) as well as mirroring for redundancy (RAID 1).The more spindles the faster the maximum read and write rates. RAID 1 alone offers no performance increase over a single spindle. Do not use RAID 5.
Hardware RAID Controller	The RAID controller should always be hardware based and server quality.
Disk Speed	15 KRPM (15K). There is a linear gain in performance with disk speed. While the cost of 15K drives is higher, this will partially be offset by the need for fewer spindles. Once a controller is at the maximum number of spindles it can manage, there is no choice but to move to a higher performance controller or swap slower drives with 15K drives.
SQL Version	SQL 2005 and 2008 are supported. 64-bit SQL should be used for all Orion installations over SL 250 or smaller Orion license levels with NetFlow.
RAM	SL 500 and below – 2GB, SL 2000, 5GB+, SLX – 20GB+. In general, the more RAM the better. For SL 2000 and above set the minimum SQL memory to eighty to ninety percent of the total server memory, but allow sufficient total memory outside this setting for the OS and any other applications.
CPU	With advances in server hardware it would be difficult to find a new server with less than 3 GHz, dual core processor. This will work for all size licenses. Do not use less than a 2 GHz processor for any Orion SQL server.

If the above server recommendations are met, the SQL server can be further optimized using several methods discussed below.

Enable Advanced Windows Extension (AWE) for 32-bit SQL.

AWE allows MS SQL enhanced access to memory under 4GB. Users enabling AWE may see dramatic improvements in MS SQL 32-bit server performance. AWE is not used for 64-bit servers. Enavling AWE depends on the version of Windows server you are running.

Separate Drives for Orion, SQL log (temp) and SQL data

Separating the temp files from the data files can help improve performance. A typical installation of this type will look like this:

- C: drive – Orion and OS. 2 drive RAID 1
- E: drive – SQL log files. 4 15K drives RAID 0
- F: drive – SQL data (Orion data). 6 15K drives RAID 10

The size of the drive space for each of the SQL drives can be determined by examining the current SQL database files.

Using RAMDisk for SQL log (temp)

RAMDisk, a third party software package, allows you to place the temp SQL files onto a logical drive that exists completely in RAM. This requires 64-bit SQL and a good amount of RAM. See the RAMDisk documentation for further requirements. This tool is very useful as it takes the most performance intensive part of SQL storage and moves them from physical disk performance levels to RAM performance levels. . Physical IOPS are measured in the hundreds where RAM IOPS are measured in the hundreds of thousands.

Solid State Drives (SSDs)

Solid state drives yield the IOP levels seen in RAM drives. These are becoming more common as the prices continue to drop. Unlike RAM drives SSDs are non volatile, so the data written to SSD remains after the system is powered down. SSDs are made of billions of microscopic transistors which store individual data bits by being in an uncharged state or a charged state. This works well for program files and data that is static. One issue with most SSDs is that the barrier that holds the charge within each transistor breaks down upon successive writes. When the transistor changes from a 1 to a 0 or a 0 to a 1 the charge state of the transistor changes and some erosion of the charge barrier occurs. After many write cycles, the transistor loses the ability to hold a charge and becomes useless for storage. SQL is very read/write intensive and so creates challenges for SSDs. New technologies aimed at maximizing SSD life span are coming onto the market. These tools can help high-end SSD arrays reach life expectations up to 8 to 10 years. Some SolarWinds' customers are using SSD and RAM arrays and report very good performance.

Orion Server Performance

Poller Performance Measurements

Much of Orion server measurement and management can be done directly from the Orion web interface. Polling managed devices for availability and performance statistics is one of the major functions of any NMS server. In order to understand how well a poller is performing, you must first understand where to look for poller performance data and how to interpret that data.

In **Settings -> Details -> Polling Engines** you to see the number of elements for each Orion poller, a breakdown of the elements by type, the actual and maximum polls per second for each data type, and the outstanding polls count for each data type. An example of this can be seen on the right. Below is a brief description of each of these metrics and an explanation of the relevance to poller performance.

Elements	297
Network Node Elements	45
Interface Elements	198
Volume Elements	54
Date Time	1/6/2011 10:23:55 AM
Paused	False
ICMP Status Polling Index	297 out of 297
SNMP Status Polling Index	297 out of 297
ICMP Status Polls per second	0
SNMP Status Polls per second	3
Max Status Polls Per Second	30
DNS Outstanding	0
ICMP Outstanding	0
SNMP Outstanding	39
ICMP Statistic Polling Index	517 out of 517
SNMP Statistic Polling Index	517 out of 517
ICMP Statistic Polls per second	0
SNMP Statistic Polls per second	14.5
Max Statistic Polls Per Second	30

Elements and Elements by Type

The total element count (Elements) shows summed element count for each poller. This example shows only one poller. This count is broken down to show the interface, node, and volume element counts. The total element count is the most important of these metrics. Typically an Orion NPM server can perform well with element counts of 8 to 9 thousand. This is known as the maximum element limit. This limit will vary with several factors including the Orion server resources, SQL server resources, SQL storage abilities, polling rates, NetFlow data rates, and the syslog message rates. The number of IP SLA operations that Orion is managing will also impact the number of maximum elements, but to a lesser degree. The Orion server resources, such as CPU, are complicated as they are comprised of several parts. The Orion server performance relies on having sufficient memory, CPU, and storage. Each of these parts has a different impact on the maximum number of elements you may be able to manage per poller. The polling increment is less complicated and has a more predictable impact on the maximum element count. The relationship between polling increment is approximately linear, so doubling the polling frequency will effectively halve the maximum element count.

Polling Indexes and Outstanding Polls

The polling indexes show the completion of the last polling cycle. The numbers indicate the total polling replies and polling requests. A disagreement in these numbers is most often associated with a network or polled device issue rather than an Orion server performance issue. The ICMP and SNMP Outstanding metrics are a measurement of the individual polls that have not received a reply. During an active polling cycle the indexes and outstanding polls numbers may be in disagreement.

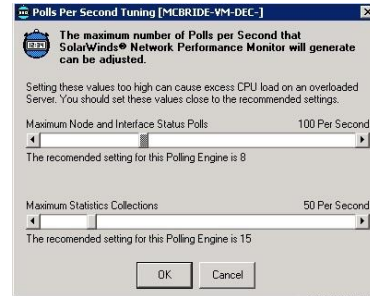
Polls per Second

The polls per second metrics show the actual poll rates and the maximum poll rates for ICMP and SNMP statistics and status polling. A well performing server will have the actual polls per second well below the maximum.

Poller Performance Tuning

If your pollers show statistics similar to the ones above, and you do not see gaps in graphs, blank reports or web slowness, there is no need to tune your pollers. If you are seeing any of these performance problems and the poller statistics show polling rates near the maximum polls per second, you will probably need to adjust the polling engine. This should only be done after the SQL server has been eliminated as the possible source of performance issues.

The Polls per Second Tuning tool, seen on the right, is used to adjust the maximum poll rate. Orion will calculate a recommended maximum poll rate based on the polling frequency and the number of elements to be polled. Restricting the maximum polling rate helps to spread the polling over time rather than slamming the network and devices with all the polls at once. The maximum possible setting is 300 polls per second.



Polling frequency is also adjustable as a global setting or granularly at the per node or per interface element settings. Global polling frequencies are set in the Orion web console under **Settings -> Polling Settings**. Since this is a global setting, it is not recommended to use the highest required polling frequency for some devices as the global setting. For example, the default node statistics polling frequency is 9 minutes. If a few of your critical servers need to be performance monitored at a 2 minute rate then the global setting should be left as is, and the individual servers polling intervals should be altered. This is done through the web console under **Settings -> Manage Nodes**. Select the nodes that need more frequent polling and click **Edit Properties** to access the nodes and change the polling.

Correcting an Overloaded Polling Engine

If polling engines are not properly managed, the result may be an overloaded engine that only becomes a problem when the web slows, gaps are seen on charts or other performance issues are seen. If you find that a polling engine is performing poorly, and determine that the poller is overloaded by the number of elements it manages and the polling intervals, here are the recommended steps to take to recover the performance.

1. Review the number of custom (UnDP) pollers you have in the system. Unnecessary UnDP pollers should be eliminated. This is accomplished in the UnDP application.
2. Examine the polling intervals and determine if the global settings have been altered to cause increased load. Remember that polling four thousand elements every five minutes is roughly equivalent to polling eight thousand elements every ten minutes. If possible, temporarily set the polling to longer intervals, especially for statistics polling. The level of adjustment depends on the severity of the performance issue and the element counts.
3. If elimination of unnecessary UnDPs and decreased polling frequency are acceptable as a long term fix, you need not go any further.
4. To return to the previous polling intervals you will have to poll fewer elements or add an additional polling server. Additional poller licenses can be purchased through SolarWinds Customer Service.
5. Once the additional poller has been installed, use the Polling Engine Load Balancing Tool in **SolarWinds Orion -> Advanced Features -> Monitor Polling Engines** to move elements to the new polling engine.
6. Set the polling intervals on the original polling engine to the rates they were set to before the issue.

Other Orion Server Performance Factors

Server Resources

While Orion does not require a lot of server resources, a lack of the minimum resources may cause performance issues. The Orion server should meet the minimums listed in the *Administrator's Guide*. Orion works well on virtual servers as long as the minimum requirements for a stand-alone server are assigned to the virtual server. The most common issue with Orion server performance is a lack of RAM memory. Allowing 4GB of memory on the Orion physical or virtual server and not sharing that server with other applications is recommended. With recent changes to the architecture of Orion, the software better takes advantage of multiple core processors. Using multi-core processors is recommended.

If the NetFlow Traffic Analyzer is used care should be taken to not overload the Orion server with an unrealistic level of flows. The number and location of NetFlow exporters should be balanced with respect to the load the flows put on the server. For more information on NetFlow rates see the NTA Administrator's Guide and [New to Networking Volume 3 – NetFlow Basics and Deployment Strategies](#).

Managing an Orion Deployment

Proactive performance management of an Orion deployment will help to detect performance issues before they impact Orion users. Besides performing the manual checks described in the previous sections, automated performance management allows the ability to be proactive. For obvious reasons, using Orion to detect Orion faults is limited but this is not true for performance management. As long as the performance thresholds are properly set, using Orion to measure Orion is not an issue. The below table shows the components of Orion discussed above and the recommended method to manage them.

Component	Management Method	Thresholds
Orion Server and Additional Polling Engines - Memory and CPU	NPM node, NPM virtual node	90 % memory used 70 % average CPU
Orion Server and Additional Polling Engines – Orion Application	Application Performance Monitor with Orion Templates	Varies with environment
SQL Server	Application Performance Monitor with SQL Templates	Varies with environment
SQL Server	NPM node	90 % memory used 90 % average CPU
Orion Web and Additional Web Servers	Application Performance Monitor or IP SLA	HTTP response according to environment and server abilities.
Shared SQL Storage	Profiler Storage Monitor	Varies with Environment

Appendix A – Other Resources

Microsoft has a very good white paper on SQL performance and IOP measurement available at <http://msdn.microsoft.com/en-us/library/cc966412.aspx>

thwack Community

The thwack community has thousands of users who regularly exchange ideas and solutions. Chances are that if you are having network management issues, there are thwack users who has already solved those issues. SolarWinds employees from Development, Support and Product Management regularly interact with SolarWinds product users on thwack, so we may also be able to help you solve your problem there too.

The thwack Content Exchange portion of thwack can be used for uploading and downloading various templates. These templates can be imported to your Orion system and customized to fit your environment.

<http://thwack.com/media/42/orion-apm-content/application-templates/29793/sql-server-via-wmi/>

<http://thwack.com/media/42/orion-apm-content/application-templates/29791/sql-server-via-snmp/>

A good discussion about measuring Orion performance can be for at this thwack thread.

<http://thwack.com/forums/48/orion-family/21/application-performance-monitor/27297/how-do-you-monitor-orion-sql-p/>