

SOLARWINDS TECHNICAL REFERENCE

APM SQL Server Template Pack

SQL Server 2005-2008 Performance (WMI)
SQL Server 2005-2008 Performance (SQL)

1
5

This document describes the templates included in the APM SQL Server Template Pack.

Copyright© 1995-2011 SolarWinds. 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 licensors. SolarWinds Orion™, SolarWinds Cirrus™, and SolarWinds Toolset™ are trademarks of SolarWinds and SolarWinds.net® and the SolarWinds logo are registered trademarks of SolarWinds All other trademarks contained in this document and in the Software are the property of their respective owners.

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 OR 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.

Microsoft® and Windows 2000® are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.

Graph Layout Toolkit and Graph Editor Toolkit © 1992 - 2001 Tom Sawyer Software, Oakland, California. All Rights Reserved.

Portions Copyright © ComponentOne, LLC 1991-2002. All Rights Reserved.

Document Revised: 04/26/2011

SQL Server 2005-2008 Performance (WMI)

This template uses Windows performance counters to assess the status and performance of Microsoft SQL Server 2005 or 2008 databases.

Prerequisites: WMI access to the target server. RPC and Remote Registry services must be running on the target server.

Credentials: Windows Administrator on the target server.

Monitored Components

Buffer Manager: Buffer cache hit ratio

This SQL Server Buffer Cache Hit Ratio counter indicates how often SQL Server goes to the buffer, not the hard disk, to get data. In OLTP applications, this ratio should exceed 90%, and ideally be over 99%. If your buffer cache hit ratio is lower than 90%, you need to go out and buy more RAM as soon as possible. If the ratio is between 90% and 99%, then you should seriously consider purchasing more RAM, as the closer you get to 99%, the faster your SQL Server will perform. In some cases, if your database is very large, you may not be able to get close to 99%, even if you put the maximum amount of RAM in your server. All you can do is add as much as you can, and then live with the consequences.

In OLAP applications, the ratio can be much less because of the nature of how OLAP works. In any case, more RAM should increase the performance of SQL Server.

Buffer Manager: Page reads/sec

The number of physical database page reads issued. 80 – 90 per second is normal; anything that is above indicates indexing or memory constraint.

Values for this counter will vary between database applications, but this information is useful when determining if SQL Server is the primary application using the disk. If the Buffer Manager page read-writes are low but disk-queue lengths are high, there might be a disk bottleneck. If the Page read-writes are higher than normal, a memory shortage is likely to exist.

Buffer Manager: Page writes/sec

The number of physical database page writes issued. 80 – 90 per second is normal; anything more and you need to check the lazy writer/sec and checkpoint counters. If these counters are also relatively high, then it is a memory constraint.

Values for this counter will vary between database applications, but this information is useful when determining if SQL Server is the primary application using the disk. If the Buffer Manager page read-writes are low but disk-queue lengths are high, there might be a disk bottleneck. If the Page read-writes are higher than normal, a memory shortage is likely to exist.

SQL Server Port

Check the availability of SQL Server Port (default is port 1433).

Note: You need to change the port value if you are using another port.

Latches: Latch waits\sec

This is the number of latch requests that could not be granted immediately. In other words, these are the amount of latches, in a one second period, that had to wait.

Latches are light-weight synchronization constructs that are designed to protect the physical integrity of a page in a similar way to how locks protect the logical consistency of rows. They are taken any time something wants to modify a page, be it moving the page from disk to memory or via versa, writing a record onto a page or changing a page's metadata.

Access Methods: Full scans\sec

This counter monitors the number of full scans on base tables or indexes. Values greater than 1 or 2 indicate that we are having table / Index page scans. If we see high CPU, then we need to investigate this counter. Otherwise, if the full scans are on small tables, we can ignore this counter. A few of the main causes of high Full Scans/sec are: missing indexes and too many rows requested.

Queries with missing indexes or too many rows requested will have a large number of logical reads and an increased CPU time.

Server work queues: Queue Length

Queue Length is the current length of the server work queue for this CPU.

A sustained queue length greater than twice the number of CPU cores might indicate processor congestion.

Note: Set a threshold according to your number of CPU cores.

Memory: Pages/sec

A page fault occurs when the operating system (OS) cannot find the requested information in its physical memory, forcing the OS to seek the information at the disk level. A soft page fault is when a page is found elsewhere in the physical memory, and a hard fault requires disk access. Most processors can handle large numbers of soft faults without significant consequence. However, hard faults, which require disk access, can cause significant delays.

This value should stay below 20-25 pages per second.

Disk: Average Disk Read Queue Length

Average Disk Read Queue Length is the average number of read requests that were queued for the selected disk during the sample interval.

The value for this counter should always be under 2. This is the most reliable counter to use when the SQL instance is using external SAN storage for its databases.

Disk: Average Disk Write Queue Length

Average Disk Write Queue Length is the average number of write requests that were queued for the selected disk during the sample interval.

The value for this counter should always be under 2. This is the most reliable counter to use when the SQL instance is using external SAN storage for its databases.

Locks: Lock Waits/sec

This counter reports how many times users waited to acquire a lock over the past second. Note that while you are actually waiting on the lock, this is not reflected in this counter. It gets incremented only when you "wake up" after waiting on the lock. If this value is nonzero, then it is an indication that there is at least some level of blocking occurring. If you combine this with the Lock Wait Time counter, you can get some idea of how long the blocking lasted. A zero value for this counter can definitively rule out blocking as a potential cause; a nonzero value will require looking at other information to determine whether it is significant.

Buffer Manager: Page life expectancy

This performance monitor counter tells you, on average, how long data pages are staying in the buffer. If this value gets below 300 seconds, this is a potential indication that your SQL Server could use more memory in order to boost performance.

Memory Manager: Total Server Memory (KB)

The Total Server Memory is the current amount of memory that SQL Server is using. If this counter is still growing, the server has not yet reached its steady-state, and it is still trying to populate the cache and get pages loaded into memory. Performance will likely be somewhat slower during this time, since more disk I/O is required at this stage. This behavior is normal. Eventually Total Server Memory should approximate Target Server Memory.

Note: Set a threshold according to your environment.

SQL Statistics: Batch Requests/sec

This counter measures the number of batch requests that SQL Server receives per second, and generally follows in step to how busy your server's CPUs are. Generally speaking, over 1000 batch requests per second indicates a very busy SQL Server, and could mean that if you are not already experiencing a CPU bottleneck, that you may be experiencing one soon. Of course, this is a relative number, and the bigger your hardware, the more batch requests per second SQL Server can handle. From a network bottleneck approach, a typical 100Mbps network card is only able to handle about 3000 batch requests per second. If you have a server that is this busy, you may need to have two or more network cards, or go to a 1Gbps network card.

Note: Sometimes low batch requests/sec can be misleading. If there were a SQL statements/sec counter, this would be a more accurate measure of the amount of SQL Server activity. For example, an application may call only a few stored procedures, yet each stored procedure does a lot of work. In that case, we will see a low number for batch requests/sec but each stored procedure (one batch) will execute many SQL statements that drive CPU and other resources. As a result, many counter thresholds based on the number of batch requests/sec will seem to identify issues because the batch requests on such a server are unusually low for the level of activity on the server.

We cannot conclude that a SQL Server is not active simply by looking at only batch requests/sec. Rather, you have to do more investigation before deciding there is no load on the server. If the average number of batch requests/sec is below 5 and other counters (such as SQL Server processor utilization) confirm the absence of significant activity, then there is not enough of a load to make any recommendations or identify issues regarding scalability.

Note: Set this threshold according to your environment.

General Statistics: User Connections

The number of users currently connected to the SQL Server.

Note: It is recommended to review this counter along with "Batch Requests/Sec". A surge in "user connections" may result in a surge of "Batch Requests/Sec". So if there is a disparity (one going up and the other staying flat or going down), then that may be a cause for concern. With a blocking problem, for example, you might see user connections, lock waits, and lock wait time all increase, while batch requests/sec decreases.

Memory Manager: Target Server Memory (KB)

Total amount of dynamic memory the server can consume.

Buffer Manager: Lazy writes/sec

How many times per second lazy writer has to flush dirty pages out of the buffer cache instead of waiting on a checkpoint. Typically dirty pages are flushed out of cache by the checkpoint process. If the value of this counter is higher than 20, then the server could use additional RAM.

Logins/sec

Total number of logins started per second. This does not include pooled connections.

Buffer Manager: Free pages

Total number of pages on all free lists.

This should be greater than 640.

Databases: Transactions/sec

The number of database transactions started in the last second.

Memory Manager: Memory Grants Pending

Total number of processes waiting to acquire a workspace memory grant.

Locks: Average Wait Time (ms)

The average wait time in milliseconds of each lock request that had a wait time.

Locks: Lock Timeouts/sec

The number of lock requests per second that time out. This number includes NOWAIT lock requests.

Databases: Data File(s) Size (KB)

The total size of the database data files.

Transactions: Longest Transaction Running Time

The length of time in seconds the transaction that has been running the longest has been active.

Log Flushes/sec

The total number of log bytes flushed.

SQL Compilations/sec

The number of SQL compilations that occur each second. Values higher than 100 indicate a high proportion of adhoc queries and may be using up valuable CPU time. Solutions include rewriting adhoc queries as stored procedures and using `sp_executeSQL`.

This value should be as low as possible.

Locks: Number of Deadlocks/sec

The number of lock requests that end up in a deadlock per second.

This value should less than 1.

Database: Log File(s) Size (KB)

The total size of the database log files.

SQL Server 2005-2008 Performance (SQL)

This template assesses the performance of a Microsoft SQL Server 2005-2008 database by retrieving performance data from the built-in master..sysperfinfo pseudo-table.

Prerequisites: None.

Credentials: SQL Server user name and password.

Monitored Components

User Connections

The number of users currently connected to the SQL Server.

Note: It is recommended to review this counter along with “Batch Requests/Sec”. A surge in “user connections” may result in a surge of “Batch Requests/Sec”. So if there is a disparity (one going up and the other staying flat or going down), then that may be a cause for concern. With a blocking problem, for example, you might see user connections, lock waits, and lock wait time all increase, while batch requests/sec decrease.

Total Server Memory (KB)

The Total Server Memory is the current amount of memory that SQL Server is using. If this counter is still growing, the server has not yet reached its steady-state, and it is still trying to populate the cache and get pages loaded into memory. Performance will likely be somewhat slower during this time since more disk I/O is required at this stage. This behavior is normal. Eventually Total Server Memory should approximate Target Server Memory.

Set a threshold according to your environment.

Target Server Memory (KB)

The total amount of dynamic memory the server can consume.

Buffer cache hit ratio

This SQL Server Buffer Cache Hit Ratio counter indicates how often SQL Server goes to the buffer, not the hard disk, to get data. In OLTP applications, this ratio should exceed 90%, and ideally be over 99%. If your buffer cache hit ratio is lower than 90%, you need to go out and buy more RAM as soon as possible. If the ratio is between 90% and 99%, then you should seriously consider purchasing more RAM, as the closer you get to 99%, the faster your SQL Server will perform. In some cases, if your database is very large, you may not be able to get close to 99%, even if you put the maximum amount of RAM in your server. All you can do is add as much as you can, and then live with the consequences.

In OLAP applications, the ratio can be much less because of the nature of how OLAP works. In any case, more RAM should increase the performance of SQL Server.

Page reads/sec

Number of physical database page reads issued. 80 – 90 per second is normal; anything that is above indicates indexing or memory constraint.

Values for this counter will vary between database applications, but this information is useful when determining if SQL Server is the primary application using the disk. If the Buffer Manager page read-writes are low but disk-queue lengths are high, there might be a disk bottleneck. If the Page read-writes are higher than normal, a memory shortage is likely to exist.

Page writes/sec

Number of physical database page writes issued. 80 – 90 per second is normal, anything more we need to check the lazy writer/sec and checkpoint counters, if these counters are also relatively high then, it is a memory constraint.

Values for this counter will vary between database applications, but this information is useful when determining if SQL Server is the primary application using the disk. If the Buffer Manager page read-writes are low but disk-queue lengths are high, there might be a disk bottleneck. If the Page read-writes are higher than normal, a memory shortage is likely to exist.

Lazy writes/sec

Number of buffers written per second by the lazy writer system process. The lazy writer flushes out old, dirty buffer cache pages to make them available for reuse. If the value of this counter is higher than 20, then the server could use additional RAM.

Page Life Expectancy

The time in seconds before the average data page is removed from the cache buffer.

If the average page life falls below 300 seconds, this may indicate that your SQL server may require more RAM to improve performance.

Lock Waits/sec

This counter reports how many times users waited to acquire a lock over the past second. Note that while you are actually waiting on the lock, this is not reflected in this counter. It gets incremented only when you “wake up” after waiting on the lock. If this value is nonzero, then it is an indication that there is at least some level of blocking occurring. If you combine this with the Lock Wait Time counter, you can get some idea of how long the blocking lasted. A zero value for this counter can definitively rule out blocking as a potential cause; a nonzero value will require looking at other information to determine whether it is significant.

Logins/sec

Total number of logins started per second. This does not include pooled connections.

Latch Waits/sec

This is the number of latch requests that could not be granted immediately. In other words, these are the amount of latches in a one second period that had to wait.

Latches are light-weight synchronization constructs that are designed to protect the physical integrity of a page in a similar way to how locks protect the logical consistency of rows. They're taken any time something wants to modify a page, be it moving the page from disk to memory or via versa, writing a record onto a page, or changing a page's metadata.

Full Scans/sec

This counter monitors the number of full scans on base tables or indexes. Values greater than 1 or 2 indicate that we are having table / Index page scans. If we see high CPU then we need to investigate this counter. Otherwise, if the full scans are on small tables, we can ignore this counter. Two of the main causes of high Full Scans/sec are missing indexes and too many rows requested.

Queries with missing indexes or too many rows requested will have a large number of logical reads and an increased CPU time.

Batch Requests/sec

This counter measures the number of batch requests that SQL Server receives per second, and generally follows in step to how busy your server's CPUs are. Generally speaking, over 1000 batch requests per second indicates a very busy SQL Server, and could mean that if you are not already experiencing a CPU bottleneck, that you may be experiencing one soon. Of course, this is a relative number, and the bigger your hardware, the more batch requests per second SQL Server can handle. From a network bottleneck approach, a typical 100Mbps network card is only able to handle about 3000 batch requests per second. If you have a server that is this busy, you may need to have two or more network cards, or go to a 1Gbps network card.

Note: Sometimes low batch requests/sec can be misleading. If there were a SQL statements/sec counter, this would be a more accurate measure of the amount of SQL Server activity. For example, an application may call only a few stored procedures, yet each stored procedure does a lot of work. In that case, we will see a low number for batch requests/sec but each stored procedure (one batch) will execute many SQL statements that drive CPU and other resources. As a result, many counter thresholds based on the number of batch requests/sec will seem to identify issues, because the batch requests on such a server are unusually low for the level of activity on the server.

We cannot conclude that a SQL Server is not active simply by looking at only batch requests/sec. Rather, you have to do more investigation before deciding there is no load on the server. If the average number of batch requests/sec is below 5 and other counters (such as SQL Server processor utilization) confirm the absence of significant activity, then there is not enough of a load to make any recommendations or identify issues regarding scalability.

Note: Set this threshold according to your environment.

Blocked Queries

The number of queries that are currently blocked. Poorly written queries can block even the faster, more efficient queries and make them slow.

This value should be as low as possible.

Fragmentation (%) of Most Fragmented Table

The fragmentation percentage of the most fragmented table in the database. Fragmentation occurs naturally because INSERT, UPDATE, and DELETE statements are not typically distributed equally among the table rows and its indexes, creating variations in how full each page is. For queries that scan portions of the indexes of a table, fragmentation can cause additional page reads.

The fragmentation percentage of a table should ideally be as close to zero as possible, but values from 0 – 10% are usually acceptable for day-to-day operations. Moderate fragmentation is 30%, severe fragmentation is 50%.

Fragmented Tables in Database

The number of tables in the database that have one or more out-of-order index pages.

Row Count of Largest Table

The number of rows in the largest table in the database.

Transactions/sec

The number of database transactions started in the last second.

Memory Grants Pending

Total number of processes waiting to acquire a workspace memory grant.

Average Lock Wait Time (ms)

The average wait time in milliseconds of each lock request that had a wait time.

This value should be kept under 500ms. Wait times over 500ms may indicate blocking.

Lock timeouts/sec

The number of lock requests per second that time out. This number includes NOWAIT lock requests.

Total Size (KB) of Database Files

The total size of the database data files.

Longest Transaction Running Time

The length of time in seconds the transaction that has been running the longest has been active.

Log Flushes/sec

The total number of log bytes flushed.

SQL Compilations/sec

The number of SQL compilations that occur each second. Values higher than 100 indicate a high proportion of adhoc queries and may be using up valuable CPU time. Solutions include rewriting adhoc queries as stored procedures and using sp_executeSQL.

This value should be as low as possible, preferably under 10% of the Batch Requests/sec.

Deadlocks/sec

The number of lock requests that end up in a deadlock per second.

This value should be less than 1.

Total Size (KB) of Log Files

The total size of the database log files.