

SQL Sentry Performance Metrics

Last Modified on 11 August 2022




This article covers the various performance metrics displayed by the **Performance Analysis Dashboard** and **Performance Analysis Overview**, and how to interpret different metric values and combinations of values across different metrics for **SQL Sentry**.

Note: For **Mode:** S = Sample and H = History.




SQL Server Metrics




Section	Metric	Description		
SQL Server Activity	User Connections	The total number of connections established to the SQL Server. Details for each connection can be viewed by querying the sys.dm_exec_connections DMV.		
		<table border="1"><tr><td>Mode:</td><td>S</td></tr><tr><td>Type:</td><td>Last value</td></tr></table>	Mode:	S
Mode:	S			
Type:	Last value			
SQL Server Activity	Blocked Processes	The total number of blocked processes detected on the SQL Server. The Dashboard is designed to give you information about the number of blocked processes. The Blocking SQL tab should be used for in depth troubleshooting of any blocking issues.		
		<table border="1"><tr><td>Mode:</td><td>S</td></tr><tr><td>Type:</td><td>Last Value</td></tr></table>	Mode:	S
Mode:	S			
Type:	Last Value			

Section	Metric	Description				
SQL Server Activity	Batches	<p>The total number of select, insert, or delete statements per second, including those inside a stored procedure. The name is somewhat misleading since it doesn't represent batches (groups of multiple statements) in the traditional sense. It's one of the best measures of overall activity on a SQL Server.</p> <p>Over 1000 Mb per second is generally considered moderate to high activity. A 100Mb network can reach saturation at around 3000 Mb per second.</p> <p>If high, use QuickTrace™ to determine which applications, hosts, etc., are responsible.</p> <table border="1" data-bbox="703 696 1406 882"> <tr> <td data-bbox="703 696 1023 779">Mode:</td> <td data-bbox="1023 696 1406 779">S, H</td> </tr> <tr> <td data-bbox="703 779 1023 882">Type:</td> <td data-bbox="1023 779 1406 882">Avg/sec</td> </tr> </table>	Mode:	S, H	Type:	Avg/sec
Mode:	S, H					
Type:	Avg/sec					
SQL Server Activity	Transactions	<p>The total number of transactions per second across all databases on the server. A transaction can be either a user-defined statement block surrounded by a BEGIN TRAN and END TRAN, or an individual DML statement (insert, update or delete).</p> <p>Compare with batches per second. On systems with high DML you typically want to see a low ratio of transactions to batches. A low ratio indicates that the individual statements are being bundled together, and can result in dramatically higher throughput and reduced IO due to log flushes.</p> <p>Transactions per second can be higher on SQL Server 2005 and 2008 due to increased system process activity.</p> <p>If high, use QuickTrace to determine which applications, hosts, etc., are responsible.</p> <table border="1" data-bbox="703 1592 1406 1756"> <tr> <td data-bbox="703 1592 1023 1675">Mode:</td> <td data-bbox="1023 1592 1406 1675">S, H</td> </tr> <tr> <td data-bbox="703 1675 1023 1756">Type:</td> <td data-bbox="1023 1675 1406 1756">Avg/sec</td> </tr> </table>	Mode:	S, H	Type:	Avg/sec
Mode:	S, H					
Type:	Avg/sec					

Section	Metric	Description						
SQL Server Activity	Compiles	<p>The total number of initial compiles and recompiles per second. The value should generally be < 10 percent of batches per second. Higher values indicate plan reuse is low, and will generally correlate with high CPU, since plan compilation is a CPU intensive operation. It may also correlate with low cache hit ratios for object and/or SQL plans (see "Plan cache hit ratios" below).</p> <p>It can also be a strong indicator of memory pressure, since there may not be enough room to keep all plans in cache.</p> <p>If you see consistently high compiles, run QuickTrace and sort results by Cache misses/sec, then expand details to view the actual queries or procedure calls that are causing the misses, which will be highlighted.</p> <p>Also see:</p> <ul style="list-style-type: none"> • Batch Compilation, Recompilation, and Plan Caching Issues in SQL Server 2005 • Caching Mechanisms <table border="1" data-bbox="703 920 1406 1189"> <tr> <td>Mode:</td> <td>S, H</td> </tr> <tr> <td>Type:</td> <td>Avg/sec</td> </tr> <tr> <td> May correlate:</td> <td>High CPU Low cache hit ratios</td> </tr> </table>	Mode:	S, H	Type:	Avg/sec	 May correlate:	High CPU Low cache hit ratios
Mode:	S, H							
Type:	Avg/sec							
 May correlate:	High CPU Low cache hit ratios							
SQL Server Activity	Recompiles	<p>The number of recompiles per second. The value should generally be < 10 percent of initial compiles per second. Also see Compiles above.</p> <table border="1" data-bbox="703 1384 1406 1547"> <tr> <td>Mode:</td> <td>S, H</td> </tr> <tr> <td>Type:</td> <td>Avg/sec</td> </tr> </table>	Mode:	S, H	Type:	Avg/sec		
Mode:	S, H							
Type:	Avg/sec							

Section	Metric	Description						
SQL Server Activity	Key lookups	<p>The number of times per second that the query processor had to perform a key lookup, across all queries. Lookups occur when the index being used is non-covering, meaning it doesn't include all of the columns required by the query. For each row returned by the index operation, the query processor has to go back to either the clustered index to perform a key lookup, or the base table to perform a RID lookup in the case of a heap. Lookups are a high overhead operation, especially when large number of rows are involved, because each lookup incurs a random I/O and additional processing. This often correlates with higher CPU usage and page reads. Lookups can be eliminated by using a covering index, adjusting joins to reduce the set so the lookup isn't needed, or using multiple indexes (intersection).</p> <p>Also see:</p> <ul style="list-style-type: none"> • SQLskills blog Kimberly L. Tripp Improving my SQL skills through your questions <table border="1" data-bbox="703 920 1406 1189"> <tr> <td data-bbox="703 920 1023 1003">Mode:</td> <td data-bbox="1023 920 1406 1003">S, H</td> </tr> <tr> <td data-bbox="703 1003 1023 1086">Type:</td> <td data-bbox="1023 1003 1406 1086">Avg/sec</td> </tr> <tr> <td data-bbox="703 1086 1023 1189">🔗 May correlate:</td> <td data-bbox="1023 1086 1406 1189">CPU Usage: User time Page Reads</td> </tr> </table>	Mode:	S, H	Type:	Avg/sec	🔗 May correlate:	CPU Usage: User time Page Reads
Mode:	S, H							
Type:	Avg/sec							
🔗 May correlate:	CPU Usage: User time Page Reads							

Section	Metric	Description						
SQL Server Activity	Forwarded records	<p>The number of times per second that the query processor had to lookup forwarded records, across all queries. Forwarded records occur in tables with no clustered index (heaps) when rows become too large to fit on the page and have to be relocated. Over time, this can cause severe fragmentation and queries to incur much higher than normal I/Os, specifically random reads. This can correlate with high SQL Server page reads, and high SQL Server disk wait time, data file and physical disk latency if the disk system isn't keeping up with the additional reads. On many systems it's not unusual for this counter to stay at zero if all tables have a clustered index, any heaps aren't fragmented, or they just aren't accessed frequently.</p> <p>Also see:</p> <ul style="list-style-type: none"> • Knowing about 'Forwarded Records' can help diagnose hard to find performance issues. • Geek City: What's Worse Than a Table Scan? <table border="1" data-bbox="703 857 1406 1249"> <tr> <td data-bbox="703 857 1023 936">Mode:</td> <td data-bbox="1023 857 1406 936">S, H</td> </tr> <tr> <td data-bbox="703 936 1023 1014">Type:</td> <td data-bbox="1023 936 1406 1014">Avg/sec</td> </tr> <tr> <td data-bbox="703 1014 1023 1249"> May correlate:</td> <td data-bbox="1023 1014 1406 1249"> SQL Server Memory: Read pages SQL Server Waits by Category: Disk Disk IO Database IO: ms/Read </td> </tr> </table>	Mode:	S, H	Type:	Avg/sec	 May correlate:	SQL Server Memory: Read pages SQL Server Waits by Category: Disk Disk IO Database IO: ms/Read
Mode:	S, H							
Type:	Avg/sec							
 May correlate:	SQL Server Memory: Read pages SQL Server Waits by Category: Disk Disk IO Database IO: ms/Read							
SQL Server Activity	Backup MB/sec	<p>The data rate in MB/sec for any backup operations taking place on the server.</p> <table border="1" data-bbox="703 1417 1406 1581"> <tr> <td data-bbox="703 1417 1023 1496">Mode:</td> <td data-bbox="1023 1417 1406 1496">S, H</td> </tr> <tr> <td data-bbox="703 1496 1023 1581">Type:</td> <td data-bbox="1023 1496 1406 1581">MB/sec</td> </tr> </table>	Mode:	S, H	Type:	MB/sec		
Mode:	S, H							
Type:	MB/sec							

Section	Metric	Description						
SQL Server Activity	Send Queue / Redo Queue	<p>The Send Queue and Redo Queue are applicable for servers involved in a database Mirroring Session.</p> <p>The Send Queue represents unsent the log that has accumulated on the log disk of the principal database. The Redo Queue is the log waiting on the mirror's disk. See also: Database Mirroring Sessions</p> <table border="1" data-bbox="707 577 1406 741"> <tr> <td data-bbox="707 577 1023 656">Mode:</td> <td data-bbox="1023 577 1406 656">S, H</td> </tr> <tr> <td data-bbox="707 656 1023 741">Type:</td> <td data-bbox="1023 656 1406 741">MB</td> </tr> </table>	Mode:	S, H	Type:	MB		
Mode:	S, H							
Type:	MB							
SQL Server Waits	CPU percent of Total Waits	<p>CPU percentage of Total Waits represents the percentage of all waits which are signal waits. <i>Signal wait</i> time is the time a thread has spent waiting on the CPU after being signaled that its resource is available. A high CPU percentage of Total Waits percentage may indicate CPU pressure.</p> <table border="1" data-bbox="707 1037 1406 1279"> <tr> <td data-bbox="707 1037 1023 1115">Mode:</td> <td data-bbox="1023 1037 1406 1115">S</td> </tr> <tr> <td data-bbox="707 1115 1023 1193">Type:</td> <td data-bbox="1023 1115 1406 1193">Percent</td> </tr> <tr> <td data-bbox="707 1193 1023 1279"> May correlate:</td> <td data-bbox="1023 1193 1406 1279">High CPU</td> </tr> </table>	Mode:	S	Type:	Percent	 May correlate:	High CPU
Mode:	S							
Type:	Percent							
 May correlate:	High CPU							

Section	Metric	Description				
SQL Server Waits	Wait Time: by Category	<p>The average wait time (in milliseconds) per second for all processes on the server, broken down by major physical resource category (Network, CPU, Memory, Disk). Although there are hundreds of wait types, only the wait types that can definitively be attributed to one of the physical resource categories are included in the calculations for this chart. The Other category is for a few other important wait types that can either affect performance in more than one major category, or cannot be directly attributed to any category with absolute certainty, such as backups and parallelism respectively.</p> <p>Waits by category is one of the most important charts on the dashboard, because it provides an instant profile of the SQL Server and where it's spending the most time waiting for physical resources. If waits by category are consistently low, then what the other dashboard charts are showing is less important. For example, if CPU and batches look unusually high, but CPU waits are low, then the server hardware is effectively handling the load.</p> <p>Total waits of less than 200ms is very good. Between 200ms and 1000ms is average. Greater than 1000ms likely requires some attention to determine where the bottleneck lies. Over 5000ms may indicate severe bottlenecking.</p> <p>The total wait time may be higher by virtue of a large number of processes (spids) active on the server, because wait time is summed across all processes, it isn't a per process average. This can be especially applicable to the Other category, because several processes experiencing parallelism at the same time can cause it to spike to very high levels.</p> <p>To view the specific wait classes involved for a particular interval, choose the interval by selecting on the chart, and then select the Sample button on the toolbar.</p> <p>If you hover over a category, the detailed wait types for that time sample are shown in a tooltip. You can also see the specific wait types for a range by running the SQL Server Wait Stats Analysis report from the Reports > Performance Analysis menu.</p> <p>See also:</p> <ul style="list-style-type: none"> • SQL Server Best Practices Article • Wait statistics, or please tell me where it hurts • What to do (or not do) about top wait stats • Adding Custom Wait Categories <table border="1" data-bbox="703 1910 1406 2072"> <tr> <td data-bbox="703 1910 1023 1989">Mode:</td> <td data-bbox="1023 1910 1406 1989">H</td> </tr> <tr> <td data-bbox="703 1989 1023 2072">Type:</td> <td data-bbox="1023 1989 1406 2072">Avg ms/sec</td> </tr> </table>	Mode:	H	Type:	Avg ms/sec
Mode:	H					
Type:	Avg ms/sec					

Section	Metric	Description				
SQL Server Waits	Wait Time: by Class	<p>The average wait time (in milliseconds) per second for all processes on the server, broken down by wait class. There are approximately 30 different wait classes, and each represents a particular SQL Server functional area or type of activity. Meaningless or innocuous wait types such as timer and queue waits are pre-filtered from view. Each class is further broken down into resource and CPU waits.</p> <p>If you hover over a class column, the detailed wait types are shown in a tooltip.</p> <table border="1" data-bbox="703 562 1407 725"> <tr> <td data-bbox="703 562 1023 645">Mode:</td> <td data-bbox="1023 562 1407 645">S</td> </tr> <tr> <td data-bbox="703 645 1023 725">Type:</td> <td data-bbox="1023 645 1407 725">Avg ms/sec</td> </tr> </table>	Mode:	S	Type:	Avg ms/sec
Mode:	S					
Type:	Avg ms/sec					
SQL Server Memory	Buffer cache size	<p>The current size of the buffer cache, in MB. You want this to be as large as possible for maximum performance, and on a dedicated SQL Server it should consume most of the SQL Server memory and physical memory.</p> <table border="1" data-bbox="703 958 1407 1122"> <tr> <td data-bbox="703 958 1023 1041">Mode:</td> <td data-bbox="1023 958 1407 1041">S, H</td> </tr> <tr> <td data-bbox="703 1041 1023 1122">Type:</td> <td data-bbox="1023 1041 1407 1122">MB</td> </tr> </table>	Mode:	S, H	Type:	MB
Mode:	S, H					
Type:	MB					
SQL Server Memory	Plan cache size (SQL)	<p>The current size of the cache used for query plans, in MB. This includes adhoc, autoparameterized, and prepared plans. A high value in proportion to the buffer cache may indicate query plans aren't being effectively reused.</p> <p>See also: Caching Mechanisms</p> <table border="1" data-bbox="703 1451 1407 1615"> <tr> <td data-bbox="703 1451 1023 1534">Mode:</td> <td data-bbox="1023 1451 1407 1534">S, H</td> </tr> <tr> <td data-bbox="703 1534 1023 1615">Type:</td> <td data-bbox="1023 1534 1407 1615">MB</td> </tr> </table>	Mode:	S, H	Type:	MB
Mode:	S, H					
Type:	MB					

Section	Metric	Description				
SQL Server Memory	Plan cache size (Objects)	<p>The current size of the cache used for object plans, in MB. This includes stored procedures, functions, and triggers. A high value in proportion to the buffer cache may indicate query plans aren't being effectively reused.</p> <p>See also: Caching Mechanisms</p> <hr/> <table border="1" data-bbox="703 510 1401 589"> <tr> <td>Mode:</td> <td>S, H</td> </tr> </table> <hr/> <table border="1" data-bbox="703 589 1401 667"> <tr> <td>Type:</td> <td>MB</td> </tr> </table>	Mode:	S, H	Type:	MB
Mode:	S, H					
Type:	MB					
SQL Server Memory	Other cache size	<p>The current size of the cache used for all other plans, in MB. This includes bound trees, extended stored procedures, temporary tables, and table variables. This cache size should be low in proportion to the other plan caches. If it goes over roughly 10 percent of the object or SQL plan size, further investigation may be needed.</p> <hr/> <table border="1" data-bbox="703 965 1401 1043"> <tr> <td>Mode:</td> <td>S, H</td> </tr> </table> <hr/> <table border="1" data-bbox="703 1043 1401 1122"> <tr> <td>Type:</td> <td>MB</td> </tr> </table>	Mode:	S, H	Type:	MB
Mode:	S, H					
Type:	MB					
SQL Server Memory	Plan cache hit ratio (SQL)	<p>The ratio of hits to lookups for the query plan cache. This value should stay above 90 percent.</p> <hr/> <table border="1" data-bbox="703 1294 1401 1373"> <tr> <td>Mode:</td> <td>S, H</td> </tr> </table> <hr/> <table border="1" data-bbox="703 1373 1401 1451"> <tr> <td>Type:</td> <td>Percent</td> </tr> </table>	Mode:	S, H	Type:	Percent
Mode:	S, H					
Type:	Percent					
SQL Server Memory	Plan cache hit ratio (Objects)	<p>The ratio of hits to lookups for the object plan cache. This value should stay above 90 percent.</p> <hr/> <table border="1" data-bbox="703 1626 1401 1704"> <tr> <td>Mode:</td> <td>S, H</td> </tr> </table> <hr/> <table border="1" data-bbox="703 1704 1401 1783"> <tr> <td>Type:</td> <td>Percent</td> </tr> </table>	Mode:	S, H	Type:	Percent
Mode:	S, H					
Type:	Percent					

Section	Metric	Description				
SQL Server Memory	In-Mem OLTP	<p>The current amount of memory in MB dedicated to In-Memory OLTP. This includes Memory-optimized tables, non-durable tables, and natively compiled T-SQL modules.</p> <hr/> <table border="1"> <tr> <td>Mode:</td> <td>S, H</td> </tr> <tr> <td>Type:</td> <td>MB</td> </tr> </table>	Mode:	S, H	Type:	MB
Mode:	S, H					
Type:	MB					
SQL Server Memory	Query Grants	<p>The current amount of memory in MB allocated to the MEMORYCLERK_SQLQERESERVATIONS clerk.</p> <p>🔗Additional Information: Memory Grants Meditation: The mysterious SQL Server memory consumer with many names (Microsoft Tech Community article).</p> <hr/> <table border="1"> <tr> <td>Mode:</td> <td>S, H</td> </tr> <tr> <td>Type:</td> <td>MB</td> </tr> </table>	Mode:	S, H	Type:	MB
Mode:	S, H					
Type:	MB					
SQL Server Memory	Stolen Buffer	<p>The amount of memory that could be used for buffer, but is consumed by another task and is not available for buffer. Stolen Buffer can be reduced by optimizing queries. Granted workspace memory can reduce the amount of free memory available and increase Stolen Buffer.</p> <hr/> <table border="1"> <tr> <td>Mode:</td> <td>S, H</td> </tr> <tr> <td>Type:</td> <td>MB</td> </tr> </table>	Mode:	S, H	Type:	MB
Mode:	S, H					
Type:	MB					
SQL Server Memory	Columnstore	<p>The current amount of memory in MB allocated to columnstore indexes. This includes both clustered and nonclustered indexes.</p> <hr/> <table border="1"> <tr> <td>Mode:</td> <td>S, H</td> </tr> <tr> <td>Type:</td> <td>MB</td> </tr> </table>	Mode:	S, H	Type:	MB
Mode:	S, H					
Type:	MB					

Section	Metric	Description						
SQL Server Memory	Page reads	<p>The average number of buffer data pages read from disk per second. Ideally, this value should be at or near zero most of the time. If it's above zero, it means that the data wasn't found in the buffer cache, and so it had to be retrieved from disk. If spikes in page reads correlate with high disk latency, the disk system may not be keeping up.</p> <p>Querying newly created temp tables will also show up as page reads, as well as activity from internal tempdb objects. This includes hash joins, hash aggregates, sort and query spool operations. This means that you can still see high paging from tempdb due to query activity, even though you aren't explicitly using temp tables.</p> <p>When page reads and page writes correlate closely, it's a strong indicator that it's related to tempdb activity, because pages are being written to disk when the objects are created, then immediately read back in to memory for use by querying operations.</p> <p>If lazy writes > zero and track closely with page reads, and page life expectancy < 600, it's a strong indicator of memory pressure, because data is being moved out of buffer to make room for new data coming in.</p> <p>Lazy writes also cause page writes, but generally much less than tempdb activity. If you see high page reads, and relatively low lazy writes and page writes, it's likely memory pressure and not tempdb activity.</p> <p>See also: Working with tempdb in SQL Server 2005</p> <table border="1" data-bbox="703 1305 1406 1574"> <tr> <td data-bbox="703 1305 1023 1384">Mode:</td> <td data-bbox="1023 1305 1406 1384">S, H</td> </tr> <tr> <td data-bbox="703 1384 1023 1462">Type:</td> <td data-bbox="1023 1384 1406 1462">Avg/sec</td> </tr> <tr> <td data-bbox="703 1462 1023 1574">↗ May correlate:</td> <td data-bbox="1023 1462 1406 1574"> Disk IO Database IO: ms/Read </td> </tr> </table>	Mode:	S, H	Type:	Avg/sec	↗ May correlate:	Disk IO Database IO: ms/Read
Mode:	S, H							
Type:	Avg/sec							
↗ May correlate:	Disk IO Database IO: ms/Read							

Section	Metric	Description						
SQL Server Memory	Page writes	<p>The average number of buffer data pages writes to disk per second.</p> <p>Page writes can be caused by checkpoints, lazy writes, and tempdb activity. To calculate the approximate amount of writes related to tempdb, for any given interval, subtract checkpoints and lazy writes from total page writes.</p> <p>If high page writes correlate with high latency, the disk system may not be keeping up.</p> <p>See Page reads for more details.</p> <table border="1" data-bbox="703 667 1406 999"> <tr> <td data-bbox="703 667 1023 745">Mode:</td> <td data-bbox="1023 667 1406 745">S, H</td> </tr> <tr> <td data-bbox="703 745 1023 824">Type:</td> <td data-bbox="1023 745 1406 824">Avg/sec</td> </tr> <tr> <td data-bbox="703 824 1023 999">🔗 May correlate:</td> <td data-bbox="1023 824 1406 999"> Disk IO Database IO: ms/Write Lazy writes Checkpoint pages </td> </tr> </table>	Mode:	S, H	Type:	Avg/sec	🔗 May correlate:	Disk IO Database IO: ms/Write Lazy writes Checkpoint pages
Mode:	S, H							
Type:	Avg/sec							
🔗 May correlate:	Disk IO Database IO: ms/Write Lazy writes Checkpoint pages							
SQL Server Memory	Page life expectancy	<p>The average lifespan of a data page. If this value is less than 600, it's an indicator of memory pressure. Ideally, it should be much higher than 600 if ample memory is available. In general, the larger the buffer cache size, the higher it should be. This is the best universal indicator of memory pressure.</p> <table border="1" data-bbox="703 1323 1406 1570"> <tr> <td data-bbox="703 1323 1023 1402">Mode:</td> <td data-bbox="1023 1323 1406 1402">S</td> </tr> <tr> <td data-bbox="703 1402 1023 1480">Type:</td> <td data-bbox="1023 1402 1406 1480">Last value</td> </tr> <tr> <td data-bbox="703 1480 1023 1570">🚨 Critical range:</td> <td data-bbox="1023 1480 1406 1570"><600</td> </tr> </table>	Mode:	S	Type:	Last value	🚨 Critical range:	<600
Mode:	S							
Type:	Last value							
🚨 Critical range:	<600							

Section	Metric	Description						
SQL Server Memory	Memory grants pending	<p>The number of processes waiting for a query workspace memory grants. Ideally, this value should be zero at all times, but it can go above zero in cases of severe memory pressure.</p> <p>When it does, RESOURCE_SEMAPHORE waits will also be > zero, since this wait is a measure of the time that queries had to wait for memory grants. This type will be visible in the Waits chart tooltips for the Memory class and category.</p> <table border="1"> <tr> <td>Mode:</td> <td>S</td> </tr> <tr> <td>Type:</td> <td>Last value</td> </tr> <tr> <td>🔔 Critical range:</td> <td>> Zero</td> </tr> </table>	Mode:	S	Type:	Last value	🔔 Critical range:	> Zero
Mode:	S							
Type:	Last value							
🔔 Critical range:	> Zero							
Database IO	Read latency by database file	<p>The average time in milliseconds each physical disk read is taking for a particular database file. The top 10 database files (data and transaction log) with the highest latency for the specified date range are shown. In History mode, select the dropdown box to determine whether total database latency or a specific database file is shown on the chart.</p> <p>See the Disk IO: Read latency section for more details and performance ranges.</p> <table border="1"> <tr> <td>Mode:</td> <td>S, H</td> </tr> <tr> <td>Type:</td> <td>Avg ms/Read</td> </tr> </table>	Mode:	S, H	Type:	Avg ms/Read		
Mode:	S, H							
Type:	Avg ms/Read							
Database IO	Write latency by database file	<p>The average time in milliseconds each physical disk write is taking for a particular database file.</p> <p>See Database IO: Read latency and Disk IO: Read latency sections above for more details.</p> <table border="1"> <tr> <td>Mode:</td> <td>S, H</td> </tr> <tr> <td>Type:</td> <td>Avg ms/Write</td> </tr> </table>	Mode:	S, H	Type:	Avg ms/Write		
Mode:	S, H							
Type:	Avg ms/Write							
		<p>The average pages per second written to disk by the checkpoint process. Checkpoints flush all dirty buffer pages for a given database to disk and are a normal part of SQL Server operations. The frequency of checkpoints and volume of checkpoint pages is dictated directly by the</p>						

Section	Metric	Description
Database IO	Checkpoint pages	<p>Recovery Interval server option. SQL Server uses checkpoints to batch writes to disk, which is generally more efficient. However, if the volume of each checkpoint is too high and you see a correlation with high disk latency, it may indicate that the disk system isn't keeping up.</p> <p>Note: SQL Server 2016 introduced Indirect Checkpoints that are defined and configured at the database level. Indirect Checkpoints are set by default for SQL Server 2016 databases.</p> <p>Checkpoints may correlate with lower page life expectancy, but only because when pages are flushed from memory PLE can drop, not because there is a direct correlation with memory pressure and checkpoints. It will also correlate directly with high SQL Server page writes. If you see a correlation with high disk latency or high SQL Server disk waits, then the disk system may not be keeping up with the checkpoint volume. By lowering the recovery interval you can increase the frequency and at the same time lower the volume of each checkpoint, which may lessen the overall impact on disk waits. Every system is different in this regard, however, so testing of different recovery intervals should be performed and the impact observed.</p> <p>This counter may, correlate with lazy writes, because it depends more on the activity profile of the server, even under memory pressure. If checkpoints are happening often enough to meet the specified recovery interval, there may not be much work for the lazywriter threads to do when they wake up.</p> <p>If the lazywriter threads are staying continually busy, and the number of dirty pages is low enough to meet the recovery interval, the checkpoint process may never have much, if any, work to do.</p> <p>On SQL Server 2008, you can use the -k startup parameter to control the max MB per second that can be written by the checkpoint process, in the event that the checkpoint process is overwhelming the disk system. This isn't a typical scenario, however, so this option should be tested carefully before use, and other possible solutions should be investigated first.</p> <p>If write latency exceeds 20ms, the checkpoint process will be automatically throttle to avoid overloading the disk. This can happen even when the disk is capable of easily sustaining higher throughput, and can lead to abnormally long checkpoints, and blocking of the log writer process.</p> <p>DBCC TRACEON (3502, 3504, -1); will log many additional details about the checkpoint process to the SQL Server error log, and can be very helpful for troubleshooting checkpoint related issues.</p> <p>See also: SQL Server I/O Basics, Chapter 2 SQL 2016 – It Just Runs Faster: Indirect Checkpoint Default CHECKPOINT (Transact-SQL)</p>

Section	Metric	<p>Changes in SQL Server 2016 Checkpoint Behavior Description</p> <table border="1"> <tr> <td data-bbox="679 226 1023 297">Mode:</td> <td data-bbox="1023 226 1406 297">S, H</td> </tr> <tr> <td data-bbox="679 297 1023 376">Type:</td> <td data-bbox="1023 297 1406 376">Avg/sec</td> </tr> <tr> <td data-bbox="679 376 1023 521"> May correlate:</td> <td data-bbox="1023 376 1406 521"> Disk IO Database IO ms/Write Lazy writes </td> </tr> </table>	Mode:	S, H	Type:	Avg/sec	May correlate:	Disk IO Database IO ms/Write Lazy writes		
Mode:	S, H									
Type:	Avg/sec									
May correlate:	Disk IO Database IO ms/Write Lazy writes									
Database IO	Lazy writes	<p>The average number of writes per second by the lazy writer. The lazy writer periodically scans the buffer and evicts pages that have low use counts in order to maintain a certain number of pages on the free list. Ideally, this value should be at or near zero most of the time. When there is no memory pressure, the lazy writer will generally leave data pages in memory, even those with low use counts. However, when pressure exists, the lazy writer will continually be working to make room for new data coming into the buffer.</p> <p>An indicator of memory pressure is ongoing lazy writes > zero with page reads/writes > zero and page life expectancy < 600.</p> <p>Also see Checkpoint pages above.</p> <table border="1"> <tr> <td data-bbox="679 1178 1023 1256">Mode:</td> <td data-bbox="1023 1178 1406 1256">S, H</td> </tr> <tr> <td data-bbox="679 1256 1023 1335">Type:</td> <td data-bbox="1023 1256 1406 1335">Avg/sec</td> </tr> <tr> <td data-bbox="679 1335 1023 1413"> Critical range:</td> <td data-bbox="1023 1335 1406 1413">>20</td> </tr> <tr> <td data-bbox="679 1413 1023 1525"> May correlate:</td> <td data-bbox="1023 1413 1406 1525"> Page life expectancy Page reads/writes </td> </tr> </table>	Mode:	S, H	Type:	Avg/sec	Critical range:	>20	May correlate:	Page life expectancy Page reads/writes
Mode:	S, H									
Type:	Avg/sec									
Critical range:	>20									
May correlate:	Page life expectancy Page reads/writes									

Section	Metric	Description				
Database IO	Log flushes	<p>Log flushes occur with every DML operation, and are a normal part of SQL Server activity. It's important to note that log writes to physical disk from updates to buffer pages happen immediately upon transaction commit, whereas writes to physical disk from the changed buffer pages is delayed until the next checkpoint occurs. It's critical that the physical disk system where the transaction log resides is fast enough to keep up with activity. If not, it can slow down all DML operations occurring in the database.</p> <p>Ideally each busy transaction log should have its own dedicated disks, so that writes can happen sequentially, which will minimize latency. If log flushes are high and latency is high for a transaction log file, then the disk system is likely underpowered for the current load.</p> <p>🔗Additional Information: Understanding log buffer flushes</p> <table border="1" data-bbox="707 857 1407 1021"> <tr> <td data-bbox="707 857 1023 943">Mode:</td> <td data-bbox="1023 857 1407 943">S, H</td> </tr> <tr> <td data-bbox="707 943 1023 1021">Type:</td> <td data-bbox="1023 943 1407 1021">Avg/sec</td> </tr> </table>	Mode:	S, H	Type:	Avg/sec
Mode:	S, H					
Type:	Avg/sec					